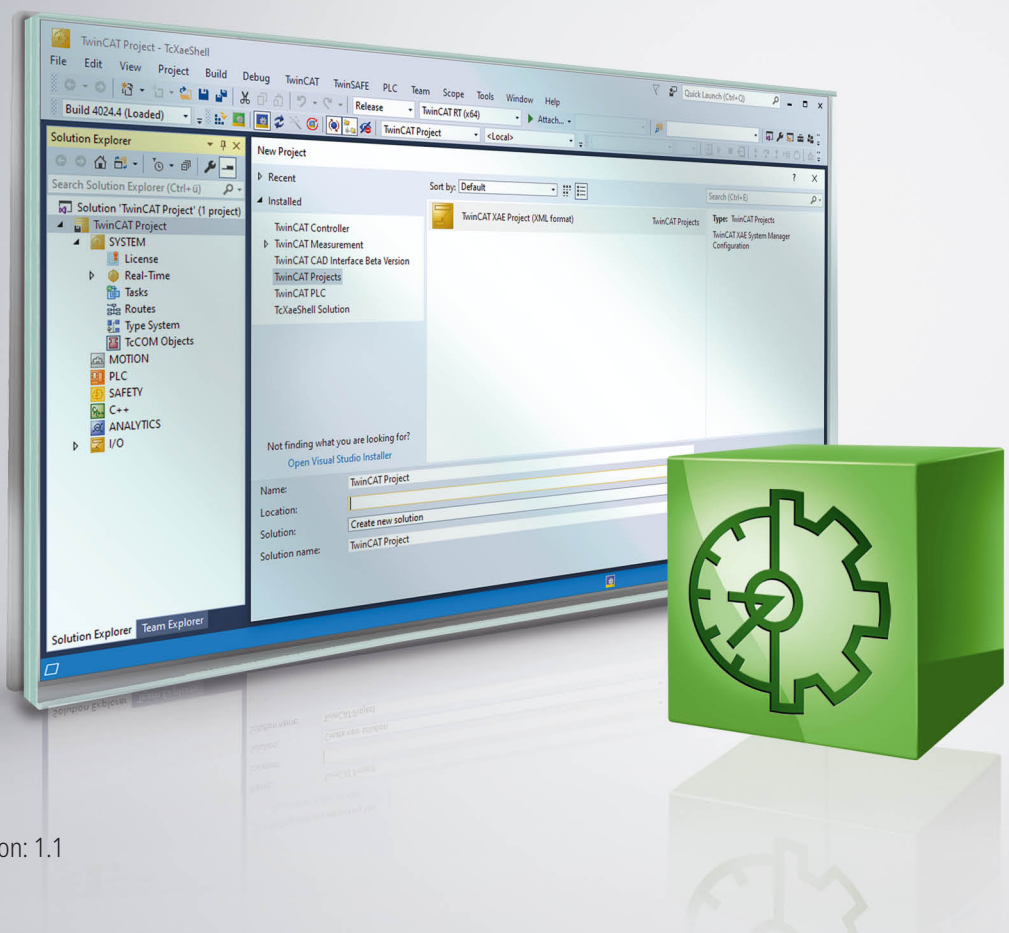
BECKHOFF New Automation Technology

Manual | EN

TC1000

TwinCAT 3 ADS .NET V5



2021-04-20 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

2 TwinCAT ADS .NET API Documentation

The TwinCAT .NET API implements support for the TwinCAT Automation Device specification (ADS). It can be used within .NET Framework programming languages and support ADS Client implementations. The ADS API interface permits:

- The Implementation of ADS Clients
- Browsing of (ADS) server side symbolic information.
- Reading and writing ProcessImage information in Raw or in type safe manner.
- Receiving ADS Notifications as events.

Getting Started

For getting started please have a look at the following documents:

- Prerequisites [▶ 16]
- Installation [▶ 16]
- Concepts [▶ 19]
- HowTo Samples [▶ 38]

Background information about the TwinCAT ADS protocol can be found here:

- ADS Introduction
- TwinCAT ADS Device concept

for common ADS information.

From the conceptual standpoint within this ADS .NET API reference documentation, the most important starting points for reading are the following Classes/Methods:

Main documentation entry points

Description	Link
Addressing ADS Devices via AmsNetId Address information.	AmsNetId [▶ 665]
Communicate to ADS Devices via the AdsClient class.	AdsClient [▶ 154]
Implement your own (virtual) ADS Server	AdsServer [▶ 1121]
Browse target system symbolic information. Creation of the SymbolLoader via Factory class.	SymbolLoaderFactory.Create(IConnection,ISymbolLoaderSettings) [▶ 1524]
Session and Connection management on top of the ADS communication channel established by the AdsClient.	AdsSession [▶ 603], AdsConnection [▶ 357]
Usage of Ads.Rx (Reactive extensions) to write reactive code.	AdsClientExtensions [▶ 1056], AdsClientExtensions [▶ 1056] The ADS reactive extensions are only available for usage, when the corresponding Nuget package is downloaded from Nuget.org. Beckhoff.TwinCAT.Ads.Reactive package on Nuget

Other Resources

HowTo Samples [▶ 38]

Version History [▶ 18]

2.1 Prerequisites

Preconditions for installing/Using the TwinCAT .NET ADS Communication API Version 5.X

To develop an application the TwinCAT .NET ADS Communication API the following software infrastructure must be available on the development system.

- An ADS/AMS Router (e.g. TwinCAT >= 4024.10) that is capable to Route the ADS/AMS Frames between systems.
- The .NET SDK to develop the application software.
- ADS Nuget packages referenced within the application project to develop TwinCAT ADS data exchange / communnication.

ADS/AMS Router

- A TwinCAT installation Version >= 4024.10.

  Because the Beckhoff.TwinCAT.Ads Version 5.X uses internal interfaces that are available only from TwinCAT 4024.10 on, an appropriate version must be installed locally.

  The package doesn't work with older installations - the actually only alternative is the AdsRouterConsole.

- Or altenativly a running instance of the 'Beckhoff.TwinCAT.Ads.TcpRouter' for customized router implementations or the 'ready-out-of-the-box' router package 'Beckhoff.TwinCAT.Ads.AdsRouterConsole' for use on system without TwinCAT.

A Software development kit for .NET (SDK)

At least one of the following SDKs and target frameworks is necessary:

- .NET 5.0 or later
- .NET Core 3.1 or later
- .NET Framework 4.61 or later
- .NET Standard 2.0 compatible SDK or later

The Beckhoff.TwinCAT.Ads components for the application.

The following Nuget packages must be added to the application project as package references:

- The 'Beckhoff.TwinCAT.Ads Nuget' package.
- and optionally the 'Beckhoff.TwinCAT.Ads.Reactive' package.

2.2 Installation

The now preferred way to install the TwinCAT ADS .NET Communication API is to use the NuGet.org package manager.

Beckhoff.TwinCAT.Ads package from Nuget.org repository.

This is the main package implementing the ADS client functionality. This is needed to establish ADS connections to local and remote devices.

Please follow the install instructions on the Nuget.org site.

Beckhoff.TwinCAT.Ads Nuget Package

Beckhoff.TwinCAT.Ads.Reactive package from Nuget.org repository.

This package installs Reactive extensions on top of the Beckhoff.TwinCAT.Ads Nuget Package and installs additional extensions to map ADSNotifications to observable events.

Please follow the install instructions on the Nuget.org site.

Beckhoff.TwinCAT.Ads.Reactive Nuget Package

3 Version History

The topics in this section describe the various changes made to the 'TwinCAT.Ads .NET API' over the life of the project.

Version History

- Version 5.0.X [▶ 18]

Other Resources

TwinCAT ADS .NET API Documentation [▶ 15]

3.1 Version 5.0.X

Feature milestones of the version 5.X.X Series

Milestones of Version 5.0.0.0

- Target support for netstandard2.0, net461, netcoreapp3.1, net50
- Asynchronous programming model (support of the async/await statements, seealso Concept of async operation [▶ 20])
- Integrated support for implementing customized AdsServers (AdsServer [▶ 1121])
- Optimized interfaces enhancing scalability and performance (see Span.T. and Memory.T.).

Other Resources

Version History [▶ 18]

4 Concepts

Concepts introduction

Concepts

Concepts discussed

Name	Description
IndexGroup / IndexOffset	Read/Write values by IndexGroup / IndexOffset [▶ 19]
Symbolic access	Read/Write values by symbolic instance path [▶ 25]
Access via symbol handle	Read/Write values by symbol handles. [▶ 26]
Asynchronous programming (async, await)	Concept of async operation [▶ 20]
ADS Notifications	Concept of ADS Notifications [▶ 22]
Symbol loader access	Access symbolic information by Symbol loader [▶ 31]
Marshalling values via ANYTYPE concept	ANY Type blittable type marshalling [▶ 27]
Dynamic automatic marshalling via Dynamic Language Runtime	Use of the .NET Framework Dynamic Language Runtime (DLR) [▶ 32]
Access via reactive extensions	Observer value changes by ADS Reactive Extensions [▶ 35]

Other Resources

TwinCAT ADS .NET API Documentation [▶ 15]

4.1 Access Data via IndexGroup/IndexOffset

Reading/Writing values by Index/Group index offset are the most basic way to access data via ADS. This address combination directly link into the process image of virtual ADS Devices.

As long the process image is static this is unproblematic and a system near access, but if the content is more dynamic and the address changes over time the IndexGroup/IndexOffset can get invalid.

Examples about moving addresses could be:

- Changed Parametrization of IO (and Re-activation)
- The PLC Online change
- New Plc Downloads

In that case other access methods could be advantageous.

Another important point is that the data access is not type safe. The values are read or written to or from byte buffers and the proper marshalling/unmarshalling is the task of the application code.

Asynchronous access

Access ProcessImage Data by IndexGroup/IndexOffset

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    UInt32 valueToRead = 0;
    UInt32 valueToWrite = 42;
```

```
    client.Connect(AmsNetId.Local, 851);

    byte[] writeData = new byte[sizeof(uint)];

    // Write an UINT32 Value
    MemoryStream writeStream = new MemoryStream(writeData);
    BinaryWriter writer = new BinaryWriter(writeStream);
    writer.Write(valueToWrite);
    ResultWrite resultWrite = await client.WriteAsync(0x4020, 0x0, writeData.AsMemory(),cancel);

    // Read an UINT32 Value
    byte[] readData = new byte[sizeof(uint)];
    ResultRead resultRead = await client.ReadAsync(0x4020, 0x0, readData.AsMemory(),cancel);

    MemoryStream readStream = new MemoryStream(readData);
    BinaryReader reader = new BinaryReader(readStream);
    valueToRead = reader.ReadUInt32();
}
```

Synchronous access

Access ProcessImage Data by IndexGroup/IndexOffset

```
using (AdsClient client = new AdsClient())
{
    UInt32 valueToRead = 0;
    UInt32 valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Write an UINT32 Value
    byte[] writeData = new byte[sizeof(uint)];
    MemoryStream writeStream = new MemoryStream(writeData);
    BinaryWriter writer = new BinaryWriter(writeStream);
    writer.Write(valueToWrite);
    client.Write(0x4020, 0x0, writeData);

    // Read an UINT32 Value
    byte[] readData = new byte[sizeof(uint)];
    int readBytes = client.Read(0x4020, 0x0, readData);

    MemoryStream readStream = new MemoryStream(readData);
    readStream.Position = 0;
    BinaryReader reader = new BinaryReader(readStream);
    valueToRead = reader.ReadUInt32();
}
```

4.2 Async programming (async, await)

Since .Version 4.0, the .NET API suppprts concurrent operation in form of the compiler support of the async/await statements. This is a special code generation supported flavor of concurrency that uses so called 'futures'. A 'future' (or promise) is a type that represents an operation that will be completed in the future and is represented by the .NET type Task or Task.TRESULT. type. This ensures that the called asynchronous method is started on call and delivers its result later on, without blocking the calling thread. As consequence the calling thread is able to process other work in the meanwhile. The deep support level in the .NET framework and the underlying code generation makes asynchronous programming almost as easy as synchronous programming.

From version 5.0.0 on, the TwinCAT.Ads API also supports the async programming model.

The main advantages are:

- Remains responsiveness of GUI applications during ADS communication
- Asynchronous programming enables scalability (mainly on the Server side)
- Easy synchronization between threads, nearly as simple as asynchronous programming.
- Functional, stateless programming style seamlessly integrating with other concurrency techniques like parallel programming (.NET Task Parallel Library) or reactive programming (Reactive Extensions [▶ 35]).

More about asynchronous programming can be read here: Asynchronous programming

and here: The Task asynchronous programming model in C#

Example

Read/Write AnyType by IndexGroup/IndexOffset (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    ResultWrite resultWrite = await client.WriteAnyAsync(0x4020, 0x0, valueToWrite,cancel);
    bool succeeded = resultWrite.Succeeded;

    ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(0x4020, 0x0, cancel);

    if(resultRead.Succeeded)
    {
    valueToRead = (uint)resultRead.Value;
    }
}
```

Read/Write AnyType by variable handle (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);

    uint valueToRead = 0;
    uint valueToWrite = 42;

    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.nCounter", cancel);
    varHandle = resultHandle.Handle;

    if (resultHandle.Succeeded)
    {
    try
    {
        ResultWrite resultWrite = await client.WriteAnyAsync(varHandle, valueToWrite, cancel);
        ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(varHandle, cancel);

        if (resultRead.Succeeded)
        valueToRead = resultRead.Value;
    }
    finally
    {
        // Unregister VarHandle after Use
        ResultAds result = await client.DeleteVariableHandleAsync(varHandle,cancel);
    }
    }
}
```

Read/Write AnyType by SymbolBrowser (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols  = await loader.GetSymbolsAsync(cancel);

    if (resultSymbols.Succeeded)
    {
    Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];
```

```
    // Works for ALL Primitive 'ANY TYPES' Symbols
    ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
    ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
        valueToRead = (uint)resultRead.Value;

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
    }
}
```

4.3 Use of ADS Notifications

If values from a PLC or NC are to be displayed continuously on a user interface, then it is very inefficient to use asynchronous read access, since this function must be called cyclically (polled triggered by a timer). Instead of using a pulling (read) model, ADS Notifications are implementing a push model. That means ADS Notifications are triggered by the sender and form a single or series of ADS messages/events. Together with these ADS Notifications, values can be transmitted. A distinction is drawn between whether the TwinCAT server is to transmit the values cyclically, or only when the values change.

In principle (raw mode) a notification is begun with the registration of the notification [M:TwinCAT.Ads.AdsClient.AddDeviceNotificationAsync(System.UInt32,System.UInt32,int,TwinCAT.Ads.NotificationSettings,System.Object, System.Threading.CancellationToken)](asynchronous) or AddDeviceNotification(synchronous). After this, events are automatically fired by TwinCAT. DeleteDeviceNotificationAsync (asynchronous) or DeleteDeviceNotification (synchronous) is used to halt the notification again. Since the number of notifications is limited, you should ensure the notifications no longer required by your program are unregistered/deleted.

There exist several 'modes' for different type of ADS Notification triggers. For a complete list please consult AdsTransMode [▶ 639].

All the following examples demonstrate how to receive ADS Notifications. The .NET ADS API supports different information layers which different levels of ADS Notification support. All are using a PLC variable in the PLC and each time the value of the PLC variable changes, an ADS Notification message is sent and the registered callback method is invoked with event arguments that contain all the necessary information (value, time stamp, ...).

Hint: Don't use time intensive executions or ADS commands inside of your callback (not more than approx. 500). Remind to sync your callback in your main thread (typically the UI thread) if necessary, because the ADS Notifications appear on a background thread.

Using ADS Notifications with Symbolic information

C#

```
private void SymbolValueChanged()
{
    using (AdsClient client = new AdsClient())
    {
    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    Symbol symbol = null;

    try
    {
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
        // DINT Type (UINT32)
        symbol = (Symbol)loader.Symbols["MAIN.nCounter"];
```

```
        // Set the Notification Settings of the Symbol if NotificationSettings.Default is not approp
riate
        // Check for change every 500 ms
        symbol.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 0);

        symbol.ValueChanged += Symbol_ValueChanged; // Registers the notification
        Thread.Sleep(5000); // Sleep the main thread to get some (asynchronous Notifications)
    }
    finally
    {
        // Unregister the Event and the underlying Handle
        symbol.ValueChanged -= Symbol_ValueChanged; // Unregisters the notification
    }
    }
}

private void Symbol_ValueChanged(object sender, ValueChangedEventArgs e)
{
    Symbol symbol = (Symbol)e.Symbol;

    // Object Value can be cast to int automatically, because it is an Primitive Value (DINT --
> Int32).
    // The Symbol information is used internally to cast the value to its appropriate .NET Type.
    int iVal = (int)e.Value;

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = iVal.ToString(), null); // Non-blocking post */
}
```

Using ADS Notifications in 'ANYTYPE' style

C#

```
//AdsStream readStream = new AdsStream(sizeof(UInt32));

private void ReceiveNotifications()
{
    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event 'Ex' handler
    client.AdsNotificationEx += Client_AdsNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    try
    {
        // Notification to a ZDINT Type (UINT32)
        // Check for change every 200 ms
        notificationHandle = client.AddDeviceNotificationEx("MAIN.nCounter",new NotificationSettings
(AdsTransMode.OnChange, 200, 0), null,typeof(uint));
        Thread.Sleep(5000); // Sleep the main thread to get some (asynchronous Notifications)
    }
    finally
    {
        // Unregister the Event / Handle
        client.DeleteDeviceNotification(notificationHandle);
        client.AdsNotificationEx -= Client_AdsNotification;
    }
    }
}

private void Client_AdsNotification(object sender, AdsNotificationExEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = (uint)e.Value;

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Asynchronous registering of Notifications

Trigger on changed values by ADS Notifications

```
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification2;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];
    int size = sizeof(UInt32);

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new Notific
ationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Synchronous registering of Notifications

Trigger on changed values by ADS Notifications

```
private void RegisterNotifications()
{
    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    try
    {
        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms

        int size = sizeof(UInt32);
        //byte[] notificationBuffer = new byte[sizeof(UInt32)];

        notificationHandle = client.AddDeviceNotification("MAIN.nCounter", size, new NotificationSet
tings(AdsTransMode.OnChange, 200, 0), null);
        Thread.Sleep(5000); // Sleep the main thread to get some (asynchronous Notifications)
    }
    finally
```

```
    {
        // Unregister the Event / Handle
        client.DeleteDeviceNotification(notificationHandle);
        client.AdsNotification -= Client_AdsNotification;
    }
    }
}

private void Client_AdsNotification(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Using reactive ADS Notifications

C#

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

4.4 Access Data via Symbolic path

The Read/Write Access by symbol path solves the issue of directly accessing the process image. With specifying the access path to the symbol, the symbol address can be found by a binary search (internally) and reading / writing symbols is independent of the location within the process image.

This access method can only be used, when the ADS device is supporting symbolic information like the TwinCAT PLC.

Because its indirect access, the performance is slightly worse than the direct access via IndexGroup/ IndexOffset. However there are internal optimizations to cache handles to the already used symbols to accelerate repeated access.

Asynchronous access

Access symbolic data by instance/symbol path

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    ResultWrite resultWrite = await client.WriteValueAsync("MAIN.nCounter", valueToWrite, cancel);
    ResultValue<uint> resultRead = await client.ReadValueAsync<uint>("MAIN.nCounter",cancel);

    if (resultRead.Succeeded)
    valueToRead = resultRead.Value;
}
```

Synchronous access

Access symbolic data by instance/symbol path

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    client.WriteValue("MAIN.nCounter", valueToWrite);
    valueToRead = (uint)client.ReadValue("MAIN.nCounter", typeof(uint));
}
```

4.5 Access Data via Symbol handles

The Read/Write Access by handle solves the issue of directly accessing the process image like the symbol path access. Because the address is accessed indirectly by the symbol path creating a variable handle, the read/write works also when the data object has changed its position within the process image.

However the cost for this are two extra ADS communication roundtrips by the 'CreateVariableHandle' and 'DeleteVariableHandle' calls compared to the IndexGroup/IndexOffset access methods. It is the responsibility of the application code to optimize these accesses.

Asynchronous access

Access symbolic values by handle

```
CancellationToken cancelT = CancellationToken.None;
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851);

    ushort valueToRead = 0; // System.UInt16
    ushort valueToWrite = 42; // System.UInt16

    // Create the Variable Handle
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.testVar", cancel); //
Test Var is defined as PLC INT

    if (resultHandle.Succeeded)
    {
    uint varHandle = 0;

    try
    {
        // Write an UINT16 Value
        byte[] writeData = new byte[sizeof(ushort)];

        MemoryStream writeStream = new MemoryStream(writeData);
        BinaryWriter writer = new BinaryWriter(writeStream);
        writer.Write(valueToWrite); // Marshal the Value
        ResultWrite resultWrite = await client.WriteAsync(varHandle, writeData.AsMemory(), cancelT);
```

```
        bool succeeded = resultWrite.Succeeded;

        // Read an UINT16 Value
        byte[] readData = new byte[sizeof(ushort)];
        ResultRead resultRead = await client.ReadAsync(varHandle, readData.AsMemory(), cancel);

        if (resultRead.Succeeded)
        {
        MemoryStream readStream = new MemoryStream(readData);
        BinaryReader reader = new BinaryReader(readStream);
        valueToRead = reader.ReadUInt16(); // Unmarshal the Value
        }
    }
    finally
    {
        // Unregister VarHandle after Use
        ResultAds result = await client.DeleteVariableHandleAsync(varHandle, cancel);
    }
    }
}
```

Synchronous access

Access symbolic values by handle

```
using (AdsClient client = new AdsClient())
{
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);
    try
    {
    UInt16 valueToRead = 0;
    UInt16 valueToWrite = 42;

    // Create the Variable Handle
    varHandle = client.CreateVariableHandle("MAIN.testVar"); //Test Var is defined as PLC INT

    // Write an UINT16 Value
    byte[] writeData = new byte[sizeof(ushort)];

    MemoryStream writeStream = new MemoryStream(writeData);
    BinaryWriter writer = new BinaryWriter(writeStream);
    writer.Write(valueToWrite); // Marshal the Value
    client.Write(varHandle, writeData.AsMemory());

    // Read an UINT16 Value
    byte[] readData = new byte[sizeof(ushort)];

    MemoryStream readStream = new MemoryStream(readData);
    client.Read(varHandle, readData.AsMemory());
    BinaryReader reader = new BinaryReader(readStream);
    valueToRead = reader.ReadUInt16(); // Unmarshal the Value
    }
    finally
    {
    // Unregister VarHandle after Use
    client.DeleteVariableHandle(varHandle);
    }
}
```

4.6 Value marshalling with ANYTYPE concept

This topic describes reading and writing variables/symbols of 'any' type with the help of the ReadAny and WriteAny (ReadSymbol, WriteSymbol) methods. The value will be marshalled / cast directly from/to its appropriate .NET type, what eases the value access.

'Any' types in this context are all types that are 'blittable' to the process image - what means that the memory layout on both sides of the data transfer is equal (e.g some primitive types) or can be marshalled by the marshalling mechanisms of .NET (see 'PlcStruct' in the example below). The memory layout specification can be customized with the 'System.Runtime.InteropServices.StructLayoutAttribute' on the .NET side (see MSDN) and the 'pack_mode' attribute on the TwinCAT PLC Side (TwinCAT 3). TwinCAT 2 only supports a memory layout of PACK = 1.

The appropriate .NET type must be known during compile time and is passed to the methods as parameter. In case of a ReadAny call, the read data will be returned as a object. The type of the object is marshalled to the type specified as parameter type. Because the data size and the memory alignment is taken from this type specification, it is so important that this specification fits to the memory representation in the ADS device (e.g. the PLC).

Because some data types(arrays and strings) need additional information, an overload of the method ReadAny exists, that takes an additional parameter args. A Full list of supported types can be found in the documentation of the overloaded method.

Reading and writing of structures

To be able to read or write PLC structures the memory layout of the structure or class in .NET must be the same as in the PLC. The layout of a structure or class can be specified with the attribute StructLayoutAttribute. The LayoutKind must be set to LayoutKind.Sequential and the pack must be set to 1

If arrays, strings or boolean values are define the class, one has to specify how these fields should be marshalled. This is accomplished with help of the MarshalAs attribute. Because arrays and strings do not have a fixed length in .NET, the property SizeConst is necessary for arrays and strings.

Marshalling values with 'ANY_TYPES' asynchronously

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851);

    // Bool value
    bool boolValue = false;
    ResultAnyValue resultBoolValue = await client.ReadValueAsync("MAIN.bool1", typeof(bool),cancel);
    boolValue = (bool)resultBoolValue.Value;
    ResultWrite resultWrite = await client.WriteValueAsync("MAIN.bool1", boolValue,cancel);

    // or
    ResultHandle resultHandleBool = await client.CreateVariableHandleAsync("MAIN.bool1",cancel); //
BOOL
    //resultHandleBool.ThrowOnError(); // or

    if (resultHandleBool.Succeeded)
    {
    ResultAnyValue resultReadBool = await client.ReadAnyAsync(resultHandleBool.Handle, typeof(bool),
 cancel);
    boolValue = (bool)resultReadBool.Value;
    ResultWrite resultWriteBool = await client.WriteAnyAsync(resultHandleBool.Handle, boolValue, can
cel);
    ResultAds resultHandleDeleteBool = await client.DeleteVariableHandleAsync(resultHandleBool.Handl
e, cancel);
    }

    // RealValue
    ResultHandle resultHandleReal = await client.CreateVariableHandleAsync("MAIN.real1",cancel); //
REAL

    if (resultHandleReal.Succeeded)
    {
    ResultAnyValue resultReadReal = await client.ReadAnyAsync(resultHandleReal.Handle, typeof(float)
, cancel); // REAL
    ResultWrite resultWriteReal = await client.WriteAnyAsync(resultHandleReal.Handle, resultReadReal
.Value, cancel);
    ResultAds resultHandleDeleteReal = await client.DeleteVariableHandleAsync(resultHandleReal.Handl
e, cancel);
    }

    // String
    ResultHandle resultHandleString = await client.CreateVariableHandleAsync("MAIN.string1",cancel);
 // STRING[80]

    if (resultHandleString.Succeeded)
    {
    ResultAnyValue resultReadString = await client.ReadAnyAsync(resultHandleString.Handle, typeof(st
ring), new int[] { 80 }, cancel); // Needs additional para for strlen
    ResultWrite resultWriteString = await client.WriteAnyAsync(resultHandleString.Handle, resultRead
String.Value, new int[] { 80 }, cancel);
    ResultAds resultHandleDeleteString = await client.DeleteVariableHandleAsync(resultHandleString.H
```

```
andle, cancel);
    }

    // ushort[]
    ResultHandle resultHandleArray = await client.CreateVariableHandleAsync("MAIN.uint1Arr", cancel)
; // ARRAY [0..9] OF UINT

    if (resultHandleArray.Succeeded)
    {
    ResultAnyValue resultReadArray = await client.ReadAnyAsync(resultHandleArray.Handle, typeof(usho
rt[]), new int[] { 10 }, cancel);
    ushort[] arrayValue = (ushort[])resultReadArray.Value;
    ResultWrite resultWriteArray = await client.WriteAnyAsync(resultHandleArray.Handle, arrayValue,
new int[] { 10 }, cancel);
    ResultAds resultHandleDeleteArray = await client.DeleteVariableHandleAsync(resultHandleArray.Han
dle, cancel);
    }

    // Complex Struct Type
    // Take care the the corresponding .NET Type is blittable / marshallable to the PLC type
    ResultHandle resultHandleStruct = await client.CreateVariableHandleAsync("MAIN.struct",cancel);

    if (resultHandleStruct.Succeeded)
    {
    ResultAnyValue resultReadStruct = await client.ReadAnyAsync(resultHandleStruct.Handle, typeof(Pl
cStruct), cancel);
    PlcStruct structValue = (PlcStruct)resultReadStruct.Value;
    ResultWrite resultWriteStruct = await client.WriteAnyAsync(resultHandleStruct.Handle, structValu
e, cancel);
    ResultAds resultHandleDeleteStruct = await client.DeleteVariableHandleAsync(resultHandleStruct.H
andle, cancel);
    }

    // ARRAY [0..9] OF STRING[80]
    // args[0] --> Number of Characters
    // args[1] --> Number of Array Elements
    // Needs additional para for strlen and number of Elements in Array
    ResultHandle resultHandleStringArray = await client.CreateVariableHandleAsync("MAIN.stringArr",c
ancel); // ARRAY [0..9] OF STRING[80]

    if (resultHandleStringArray.Succeeded)
    {
    ResultAnyValue resultReadStringArray = await client.ReadAnyAsync(resultHandleStringArray.Handle,
 typeof(string[]), new int[] { 80, 10 }, cancel);
    string[] stringArrValue = (string[])resultReadStringArray.Value;
    ResultWrite resultWriteStringArray = await client.WriteAnyAsync(resultHandleStringArray.Handle,
stringArrValue, new int[] { 80, 10 }, cancel);
    ResultAds resultHandleDeleteStringArray = await client.DeleteVariableHandleAsync(resultHandleStr
ingArray.Handle, cancel);
    }

}
```

Defining Memory layout of struct type.

```
// Attention: Dependent of the System where the PLC runs, the StructLayout of the exchanged
// Structures must match. With the ANY_TYPE concept this is realized with 'blittable' objects,
// that match on .NET and PLC side.

// Default Pack Modes:

// TC3 I64/x86: Normal, in this case Pack = 8
// TC2 x86:     Pack = 1

// On TC3 PLC side we can force the packing of structures with the attribute
// {attribute 'pack_mode' := '1'}, see also 'pack_mode' attribute in Beckhoff InfoSystem
// For TC2 is the Pack setting Pack = 1 the only possible way, because it is not selectable.

// We have to ensure that the pack mode on both sides is equal!

[StructLayout(LayoutKind.Sequential, Pack = 1, CharSet = CharSet.Ansi)]
public struct PlcStruct
{
    // Type must be 'blittable' to the corresponding PLC Struct Type
    // See MSDN for MarshalAs and Default Marshalling.

    [MarshalAs(UnmanagedType.I1)]
    public bool boolVal; // BOOL
    public byte byteVal; // BYTE
```

```
    public ushort ushortVal; // UINT
    public short shortVal; // INT
    public uint uintVal; // UDINT
    public int dintVal; // DINT
    public uint udintVal; // UDINT
    public float realVal; // REAL
    public double lrealVal; // LREAL
    [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 81)]
    public string stringVal; // STRING[80]

    [MarshalAs(UnmanagedType.U4)]
    public uint timeVal; // TIME
    [MarshalAs(UnmanagedType.U4)]
    public uint todVal; // TOD
    [MarshalAs(UnmanagedType.U4)]
    public uint dateval; // DATE
    [MarshalAs(UnmanagedType.U4)]
    public uint dtVal; // DT
}
```

Marshalling values with 'ANY_TYPES' (synchronous)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851);

    // Bool value
    bool boolValue = (bool)client.ReadValue("MAIN.bool1", typeof(bool));
    client.WriteValue("MAIN.bool1", boolValue);

    // or
    uint handleBool = client.CreateVariableHandle("MAIN.bool1"); // BOOL
    boolValue = (bool)client.ReadAny(handleBool, typeof(bool));
    client.WriteAny(handleBool, boolValue);
    client.DeleteVariableHandle(handleBool);

    // RealValue
    uint handleReal = client.CreateVariableHandle("MAIN.real1"); // REAL
    float realValue = (float)client.ReadAny(handleReal, typeof(float));
    client.WriteAny(handleReal, realValue);
    client.DeleteVariableHandle(handleReal);

    // String
    uint handleString = client.CreateVariableHandle("MAIN.string1"); // STRING[80]
    string stringValue = (string)client.ReadAny(handleString, typeof(string), new int[] { 80 }); //
Needs additional para for strlen
    client.WriteAny(handleString, stringValue, new int[] { 80 });
    client.DeleteVariableHandle(handleString);

    // ushort[]
    uint handleArray = client.CreateVariableHandle("MAIN.uint1Arr"); // ARRAY [0..9] OF UINT
    ushort[] arrayValue = (ushort[])client.ReadAny(handleArray, typeof(ushort[]), new int[] { 10 });
    client.WriteAny(handleArray, arrayValue, new int[] { 10 });
    client.DeleteVariableHandle(handleArray);

    // Complex Struct Type
    // Take care the the corresponding .NET Type is blittable / marshallable to the PLC type
    uint handleStruct = client.CreateVariableHandle("MAIN.struct");
    PlcStruct structValue = (PlcStruct)client.ReadAny(handleStruct, typeof(PlcStruct));
    client.WriteAny(handleStruct, structValue);
    client.DeleteVariableHandle(handleStruct);

    // ARRAY [0..9] OF STRING[80]
    // args[0] --> Number of Characters
    // args[1] --> Number of Array Elements
    // Needs additional para for strlen and number of Elements in Array
    uint handleStringArr = client.CreateVariableHandle("MAIN.stringArr"); // ARRAY [0..9] OF STRING[
80]
    string[] stringArr = (string[])client.ReadAny(handleStringArr, typeof(string[]), new int[] { 80,
 10 });
    client.WriteAny(handleStringArr, stringArr, new int[] { 80, 10 });
    client.DeleteVariableHandle(handleStringArr);

}
```

ADS Notifications with Type marshalling (AdsNotificationEx)

The method AddDeviceNotificationEx is used to register notifications for a PLC variable. If the value of a variable changes the event AdsNotificationEx is fired. The difference to the event AdsNotification, is that the value of the variable is stored in an object instead of in an AdsStream. Therefore one has to pass the type of the object to the method AddDeviceNotificationEx

Notifications with 'ANY_TYPES' (asynchronous)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new Notifica
tionSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint),null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle,ca
ncel); // Unregister Event
}
```

Notifications with 'ANY_TYPES'

```
private void Client_AdsNotificationEx(object sender, AdsNotificationExEventArgs e)
{
    uint value = (uint)e.Value; // Marshalled value as .NET Type
}
```

Notifications with 'ANY_TYPES' (synchronous)

```
using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    uint notificationHandle = client.AddDeviceNotificationEx("MAIN.udint", new NotificationSettings(
AdsTransMode.OnChange, 200, 200), null, typeof(uint));
    Thread.Sleep(5000); // ...
    client.DeleteDeviceNotification(notificationHandle); // Unregister Event
}
```

4.7 Access Data via Symbol Loader

Some ADS Devices (e.g. the TwinCAT PLC) provide symbolic information for download. That means all visible Symbols and DataTypes can be retrieved from the target system. While this need an extra effort to upload and hold the data, this feature helps to remove the dependency of the code/configuration running on the target device.

E.g. because the symbolic information can now be browsed and determined during runtime, the application can be written without knowing what's running on the target system. Even more having the Symbol information cached, the access of the process image data will be easier because the datasize and access (instance path) is stored in the symbol.

Dependent how it is parametrized, the symbol loader can work with 'ANY_TYPES' (marshallable Primitive types, Value marshalling with ANYTYPE concept [▶ 27]) or full dynamic symbols (Automatic dynamic marshalling of values [▶ 32]).

Example

Accessing symbolic data by preloaded Symbolic information (asynchronous)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
```

```
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols  = await loader.GetSymbolsAsync(cancel);

    if (resultSymbols.Succeeded)
    {
    Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
    ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
        valueToRead = (uint)resultRead.Value;

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
    }
}
```

Accessing symbolic data by preloaded Symbolic information (synchronous)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    Symbol symbol = (Symbol)loader.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    symbol.WriteValue(valueToWrite);
    valueToRead = (uint)symbol.ReadValue();

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: loader.Symbols, recurse: true, predicate:
filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
    Console.WriteLine(filteredSymbol.InstancePath);
    }
}
```

4.8 Automatic dynamic marshalling of values

The 'Dynamic Symbol Loader' of the .NET ADS Communication API makes use of the .NET dynamic language runtime (DLR). The dynamic language runtime is a runtime environment that adds a set of services for dynamic languages to the common language runtime (CLR). The DLR makes it easier to develop dynamic languages to run on the .NET Framework and to add dynamic features to statically typed languages.

Dynamic languages can identify the type of an object at run time, whereas in statically typed languages (without using the 'dynamic' keyword, specify object types at design time. The advantage here is - from the moment on the symbolic (and dataType) information is available from the ADS Device - Symbol/Variable values can be marshalled 'on-the-fly' during runtime in a type-safe manner.

This works not only with primitive types but also with complex types. This reduces the complexity of the written application code to access the values, because neither the type of the data must be known, nor how the value data must be marshalled from/to the process image. The price to be paid is simply that the full symbolic information and data types must be downloaded from the ADS Device by the symbol loader.

Example

Automatic marshalling values with 'Dynamic Values' (asynchronous)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoa
derSettings.DefaultDynamic);

    //dynamic symbols = loader.SymbolsDynamic;
    ResultDynamicSymbols resultSymbols = await loader.GetDynamicSymbolsAsync(cancel);

    if (resultSymbols.Succeeded)
    {
    dynamic symbols = resultSymbols.Symbols;
    dynamic main = symbols.Main;

    // Use typed object to use InfoTips
    DynamicSymbol nCounter = main.nCounter; // UDINT

    // or to be fullDynamic
    dynamic nCounter2 = main.nCounter;

    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    ResultReadValueAccess resultRead = await nCounter.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
    {
        valueToRead = (uint)resultRead.Value;

        // or
        var varValue = resultRead.Value;
        // or
        dynamic dynValue = resultRead.Value;
    }
    // Same for writing
    ResultWriteAccess resultWrite = await nCounter.WriteValueAsync(valueToWrite,cancel);

    // Or Notifications / Events (typed dynamically)
    nCounter.ValueChanged += NCounter_ValueChanged;

    //Reading complexTypes e.g. Struct

    DynamicSymbol myStructSymbol = main.plcStruct; // Dynamically created
    ResultReadValueAccess resultRead2 = await myStructSymbol.ReadValueAsync(cancel); // Takes an ADS
 Snapshot of the value

    if (resultRead2.Succeeded)
    {
        dynamic myStructVal = resultRead2.Value;

        dynamic int1Val = myStructVal.int1; // Value to an INT (short)
        dynamic valueNestedStruct = myStructVal.nestedStruct; //
value to another complex type (here a nested Struct)
    }
    myStructSymbol.ValueChanged += MyStructSymbol_ValueChanged;
    //wait 5 seconds to get some events
```

```
    Thread.Sleep(5000);
    }
}
```

Automatic marshalling values with 'Dynamic Values' (handler)

```
private void NCounter_ValueChanged(object sender, ValueChangedEventArgs e)
{
    var uintVal = e.Value;
}

private void MyStructSymbol_ValueChanged(object sender, ValueChangedEventArgs e)
{
    dynamic structValue = e.Value; // Snapshot of the whole Struct and all its contents
}
```

Automatic marshalling values with 'Dynamic Values' (synchronous)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoa
derSettings.DefaultDynamic);

    dynamic symbols = loader.SymbolsDynamic;
    dynamic main = symbols.Main;

    // Use typed object to use InfoTips
    DynamicSymbol nCounter = main.nCounter; // UDINT

    // or to be fullDynamic
    //dynamic nCounter = main.nCounter;

    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    valueToRead = (uint)nCounter.ReadValue();
    // or
    var varValue = nCounter.ReadValue();
    // or
    dynamic dynValue = nCounter.ReadValue();

    // Same for writing
    nCounter.WriteValue(valueToWrite);

    // Or Notifications / Events
    nCounter.ValueChanged += new EventHandler<ValueChangedEventArgs>(NCounter_ValueChanged);

    //Reading complexTypes e.g. Struct

    DynamicSymbol myStructSymbol = main.plcStruct; // Dynamically created
    dynamic myStructVal = myStructSymbol.ReadValue(); // Takes an ADS Snapshot of the value

    dynamic int1Val = myStructVal.int1; // Value to an INT (short)
    dynamic valueNestedStruct = myStructVal.nestedStruct; //
value to another complex type (here a nested Struct)

    myStructSymbol.ValueChanged += new EventHandler<ValueChangedEventArgs>(MyStructSymbol_ValueChang
ed);
    //wait for notifications for 5 seconds
    Thread.Sleep(5000);
}
```

Calling 'ReadValue'/'ReadValueAsync' or the incoming 'ValueChanged' notification takes a full snapshot (with snapshot time) of the value. That means, when for example subelements of a struct value will be accessed, all subvalues will represent the value of that snapshot time consistently. The starting point (or the instance that caches the consistent data) is always the 'DynamicSymbol' object that called 'ReadValue'.

An update of the value can be done directly on the value with 'UpdateValue', or with reading a new Value on the 'DynamicSymbol' ('ReadValue').

The 'ValueChanged' event on the 'DynamicSymbol' assigns a Notification for just this symbol. The 'ValueChanged' handler will contain the value completely marshalled as dynamic object.

4.9 Reactive Extensions and ADS

The Reactive Extensions (Rx) is a .NET library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers.

In ADS terms, not only the reading and writing data or symbol values can be put into reactive data streams, also ADS Notifications are a perfect fit for reactive code. This eases not only data binding to reactive frameworks (e.g. reactive UI) but also supports enhanced data manipulation via synchronous and asynchronous observers. Multithreaded and parallelized code paths that support multiple CPU cores can be written very easily without the burden of deadlock and synchronization issues.

More about .NET reactive extensions can be read here: Reactive extensions project site.

The TwinCAT ADS Reactive extensions are available via a supplement Nuget Package: Beckhoff.TwinCAT.Ads.Reactive Nuget Package

Example

Observe for Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    }
    );

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Observe for Symbol Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);
```

```
    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Observer for dynamic Symbol Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings
.DefaultDynamic);
    dynamic symbols = symbolLoader.SymbolsDynamic;
    dynamic cycleCount = symbols.TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    // Value objects can be dynamically (on the fly) created objects here (e.g. structs)
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    // We have to give the 'hint' about IValueSymbol here, that the CLR finds the Extension Method '
WhenValueChanged' during runtime.
    IDisposable subscription = ((IValueSymbol)cycleCount).WhenValueChanged().Take(20).Subscribe(valu
eObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Polling observer

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Writing values with observable subject

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
```

```
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

5 HowTo Samples

The topics in this section describe the various changes made to the 'TwinCAT.Ads .NET API' over the life of the project.

HowTo

How to use the .NET TwinCAT API

Sample	Description
Reading/writing (simple) values from/to ADS Servers.	Read/Write values [▶ 38]
Reading/writing string values.	Read/Write string values [▶ 42]
Event driven reading (ADS Notifications)	Event driven reading [▶ 49]
Read/Write data from/to ADS Servers using reactive extensions	Reactive Read/Write [▶ 50]
Upgrading existing ADS Application code (Version 4.X --> 5.X)	Upgrade 4.X --> 5.X [▶ 52]

Other Resources

TwinCAT ADS .NET API Documentation [▶ 15]

5.1 Read/Write primitive values

Reading writing Values from ADS Devices is the most essential part of the communication API. There are several options for communication with your application.

- Accessing by IndexGroup / IndexOffset
- Symbolic access by instance path and optionally use handles for the symbol
- Holding the overall Symbolic information in the SymbolLoader and use easy access via symbol objects.
- Usage the symbolic interface ITcAdsSymbol when complete SymbolLoading by SymbolLoaders is not appropriate.
- Reading / Writing values as .NET managed Types (primitive types or compound primitive types called ANY_TYPES), or complex dynamic types typesafe generated at runtime.

The following section shows the different scenarios as code snippets.

HowTo Read/Write Values

Read/Write AnyType by IndexGroup/IndexOffset (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    ResultWrite resultWrite = await client.WriteAnyAsync(0x4020, 0x0, valueToWrite,cancel);
    bool succeeded = resultWrite.Succeeded;

    ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(0x4020, 0x0, cancel);

    if(resultRead.Succeeded)
```

```
    {
    valueToRead = (uint)resultRead.Value;
    }
}
```

Read/Write AnyType by IndexGroup/IndexOffset (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    adsClient.WriteAny(0x4020, 0x0, valueToWrite);
    valueToRead = (uint)adsClient.ReadAny(0x4020, 0x0, typeof(uint));
}
```

Read/Write AnyType by variable handle (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);

    uint valueToRead = 0;
    uint valueToWrite = 42;

    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.nCounter", cancel);
    varHandle = resultHandle.Handle;

    if (resultHandle.Succeeded)
    {
    try
    {
        ResultWrite resultWrite = await client.WriteAnyAsync(varHandle, valueToWrite, cancel);
        ResultValue<uint> resultRead = await client.ReadAnyAsync<uint>(varHandle, cancel);

        if (resultRead.Succeeded)
        valueToRead = resultRead.Value;
    }
    finally
    {
        // Unregister VarHandle after Use
        ResultAds result = await client.DeleteVariableHandleAsync(varHandle,cancel);
    }
    }
}
```

Read/Write AnyType by variable handle (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint varHandle = 0;
    client.Connect(AmsNetId.Local, 851);
    try
    {
    uint valueToRead = 0;
    uint valueToWrite = 42;

    varHandle = client.CreateVariableHandle("MAIN.nCounter");
    adsClient.WriteAny(varHandle, valueToWrite);
    valueToRead = (uint)adsClient.ReadAny(varHandle, typeof(uint));
    }
    finally
    {
    // Unregister VarHandle after Use
    client.DeleteVariableHandle(varHandle);
    }
}
```

Read/Write AnyType by instance/symbol path (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;
    uint valueToRead = 0;
    uint valueToWrite = 42;
```

```
    client.Connect(AmsNetId.Local, 851);

    ResultWrite resultWrite = await client.WriteValueAsync("MAIN.nCounter", valueToWrite, cancel);
    ResultValue<uint> resultRead = await client.ReadValueAsync<uint>("MAIN.nCounter",cancel);

    if (resultRead.Succeeded)
    valueToRead = resultRead.Value;
}
```

Read/Write AnyType by instance/symbol path (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);
    client.WriteValue("MAIN.nCounter", valueToWrite);
    valueToRead = (uint)client.ReadValue("MAIN.nCounter", typeof(uint));
}
```

Read/Write AnyType by IAdsSymbol (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    ResultValue<IAdsSymbol> resultSymbol = await client.ReadSymbolAsync("MAIN.nCounter",cancel);

    if (resultSymbol.Succeeded)
    {
    ResultWrite resultWrite = await client.WriteValueAsync(resultSymbol.Value, valueToWrite, cancel)
;
    bool succeeded = resultWrite.Succeeded;

    ResultValue<uint> resultValue = await client.ReadValueAsync<uint>(resultSymbol.Value,cancel);

    if (resultValue.Succeeded)
        valueToRead = resultValue.Value;
    }
}
```

Read/Write AnyType by IAdsSymbol (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    IAdsSymbol symbol = adsClient.ReadSymbol("MAIN.nCounter");
    adsClient.WriteValue(symbol, valueToWrite);
    valueToRead = (uint)adsClient.ReadValue(symbol);
}
```

Read/Write AnyType by SymbolBrowser (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols  = await loader.GetSymbolsAsync(cancel);
```

```
    if (resultSymbols.Succeeded)
    {
    Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
    ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
        valueToRead = (uint)resultRead.Value;

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
    }
}
```

Read/Write AnyType by SymbolBrowser (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    Symbol symbol = (Symbol)loader.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    symbol.WriteValue(valueToWrite);
    valueToRead = (uint)symbol.ReadValue();

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: loader.Symbols, recurse: true, predicate:
filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
    Console.WriteLine(filteredSymbol.InstancePath);
    }
}
```

Read/Write dynamic types by SymbolBrowser (asynchronously)

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoa
derSettings.DefaultDynamic);

    //dynamic symbols = loader.SymbolsDynamic;
    ResultDynamicSymbols resultSymbols = await loader.GetDynamicSymbolsAsync(cancel);
    if (resultSymbols.Succeeded)
    {
    dynamic symbols = resultSymbols.Symbols; // Symbols collection with 'dynamic' access.
    dynamic main = symbols.Main; // Get the 'dynamic' main FB as Property
```

```
    // Use typed object to use InfoTips instead of 'dynamic'
    DynamicSymbol nCounter = main.nCounter;
    // or to be fullDynamic
    dynamic nCounter2 = main.nCounter;

    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    ResultWriteAccess resultWrite = await nCounter.WriteValueAsync(valueToWrite,cancel);
    ResultReadValueAccess resultRead = await nCounter.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
    {
        // Because the PLC value is defined as UDINT (32-Bit)
        // We get back an already Marshalled UInt32 here ...
        valueToRead = (uint)resultRead.Value;
    }
    }
}
```

Read/Write dynamic types by SymbolBrowser (synchronously)

```
using (AdsClient client = new AdsClient())
{
    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    // Primitive Parts will be automatically resolved to .NET Primitive types.
    IDynamicSymbolLoader loader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoa
derSettings.DefaultDynamic);

    dynamic symbols = loader.SymbolsDynamic;
    dynamic main = symbols.Main;

    // Use typed object to use InfoTips
    DynamicSymbol nCounter = main.nCounter;

    // or to be fullDynamic
    //dynamic nCounter = main.nCounter;

    // Works for ALL sorts of types (not restricted to ANY_TYPE basing primitive types)
    nCounter.WriteValue(valueToWrite);
    valueToRead = (uint)nCounter.ReadValue();
}
```

5.2 Read/Write string types

ADS Server usually support strings in 2 flavors. The Default (ANSI) and the Unicode encoding (STRING vs. WSTRING) The ANSI encoding reserves 1 byte per character. Unicode reserves 2.

The strings are of fixed size and therefore the length of the the reserved space within the process image is important.

HowTo Read/Write string values

Reading writing ANSI Streams:

Read/Write ANSI Strings (async)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)

    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.string",cancel); // Sym
bol "string" in MAIN defined as string

    if (resultHandle.Succeeded)
    {
    try
    {
```

```
        // Read ANSI String string[80]
        int byteSize = 81; // Size of 80 ANSI chars + /0 (STRING[80])
        PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(StringMarshaler.DefaultEncodin
g);
        byte[] buffer = new byte[byteSize];

        ResultRead resultRead = await client.ReadAsync(resultHandle.Handle, buffer.AsMemory(), cance
l);

        if (resultRead.Succeeded)
        {
        string value = null;
        converter.Unmarshal<string>(buffer.AsSpan(), out value);

        byte[] writeBuffer = new byte[byteSize];
        // Write ANSI String string[80]
        value = "Changed";
        converter.Marshal(value, writeBuffer);
        ResultWrite resultWrite = await client.WriteAsync(resultHandle.Handle, writeBuffer, cancel);
        }
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandle.Handle, cancel);
    }
    }
}
```

Read/Write ANSI Strings (sync)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    uint handle = client.CreateVariableHandle("MAIN.string"); // Symbol "string" in MAIN defined as
string

    try
    {
    // Read ANSI String string[80]
    int byteSize = 81; // Size of 80 ANSI chars + /0 (STRING[80])
    PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(StringMarshaler.DefaultEncoding);
    byte[] buffer = new byte[byteSize];

    int readBytes = client.Read(handle, buffer.AsMemory());

    string value = null;
    converter.Unmarshal<string>(buffer.AsSpan(), out value);

    // Write ANSI String string[80]
    byte[] writeBuffer = new byte[byteSize];
    value = "Changed";
    converter.Marshal(value, writeBuffer);
    client.Write(handle, writeBuffer);
    }
    finally
    {
    client.DeleteVariableHandle(handle);
    }
}
```

Reading writing UNICODE Streams:

Read/Write Unicode Strings (async)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    ResultHandle resultHandle = await client.CreateVariableHandleAsync("MAIN.wstring",cancel); // Sy
mbol "wstring" defined in MAIN as WSTRING

    if (resultHandle.Succeeded)
    {
    try
    {
        // Read UNICODE String wstring[80]
        PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(Encoding.Unicode);
        int byteSize = converter.MarshalSize(80); // Size of 80 UNICODE chars + /0 (WSTRING[80]) (16
```

```
2)
        byte[] readBuffer = new byte[byteSize];

        ResultRead resultRead = await client.ReadAsync(resultHandle.Handle, readBuffer, cancel);

        if (resultRead.Succeeded)
        {
        string value = null;
        converter.Unmarshal(readBuffer.AsSpan(), out value);

        // Write Unicode String string[80]
        value = "Changed";
        byte[] writeBuffer = new byte[byteSize];
        converter.Marshal(value, writeBuffer.AsSpan());

        ResultWrite resultWrite = await client.WriteAsync(resultHandle.Handle, writeBuffer.AsMemory(
), cancel);
        }
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandle.Handle, cancel);
    }
    }
}
```

Read/Write Unicode Strings (sync)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)
    uint handle = client.CreateVariableHandle("MAIN.wstring"); // Symbol "wstring" defined in MAIN a
s WSTRING

    try
    {
    // Read UNICODE String wstring[80]

    PrimitiveTypeMarshaler converter = new PrimitiveTypeMarshaler(Encoding.Unicode);
    int byteSize = converter.MarshalSize(80); // Size of 80 UNICODE chars + /0 (WSTRING[80]) (162)
    byte[] readBuffer = new byte[byteSize];

    int readBytes = client.Read(handle, readBuffer);

    string value = null;
    converter.Unmarshal(readBuffer.AsSpan(), out value);

    // Write Unicode String string[80]
    value = "Changed";
    byte[] writeBuffer = new byte[byteSize];
    converter.Marshal(value, writeBuffer.AsSpan());

    client.Write(handle, writeBuffer.AsMemory());
    }
    finally
    {
    client.DeleteVariableHandle(handle);
    }
}
```

Reading writing strings with ReadAny/WriteAny group of methods:

Read/Write Anystring (async)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)

    ResultHandle resultHandleStr = await client.CreateVariableHandleAsync("MAIN.string",cancel); //
Symbol "string" defined in MAIN as STRING
    ResultHandle resultHandleWStr = await client.CreateVariableHandleAsync("MAIN.wstring",cancel); /
/ Symbol "string" defined in MAIN as WSTRING

    if (resultHandleStr.Succeeded && resultHandleWStr.Succeeded)
    {
    try
    {
```

```
        ResultAnyValue resultReadStr = await client.ReadAnyStringAsync(resultHandleStr.Handle, 80, S
tringMarshaler.DefaultEncoding, cancel);
        ResultAnyValue resultReadWStr = await client.ReadAnyStringAsync(resultHandleWStr.Handle, 80,
 Encoding.Unicode, cancel);

        string changedValue = "Changed";

        // Attention, take care that the memory of the string in the process image is not exceeded!
        ResultWrite resultWriteStr = await client.WriteAnyStringAsync(resultHandleStr.Handle, change
dValue, 80, StringMarshaler.DefaultEncoding, cancel);
        ResultWrite resultWriteWStr = await client.WriteAnyStringAsync(resultHandleWStr.Handle, chan
gedValue, 80, Encoding.Unicode, cancel);
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleStr.Handle, cancel);
        ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleWStr.Handle, cancel);
    }
    }
}
```

Read/Write Anystring (sync)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local port 851 (PLC)

    uint stringHandle = client.CreateVariableHandle("MAIN.string"); // Symbol "string" defined in MA
IN as STRING
    uint wStringHandle = client.CreateVariableHandle("MAIN.wstring"); // Symbol "string" defined in
MAIN as WSTRING

    try
    {
    string str = client.ReadAnyString(stringHandle, 80, StringMarshaler.DefaultEncoding);
    string wStr = client.ReadAnyString(wStringHandle, 80, Encoding.Unicode);

    string changedValue = "Changed";

    // Attention, take care that the memory of the string in the process image is not exceeded!
    client.WriteAnyString(stringHandle, changedValue, 80, StringMarshaler.DefaultEncoding);
    client.WriteAnyString(wStringHandle, changedValue, 80, Encoding.Unicode);
    }
    finally
    {
    client.DeleteVariableHandle(stringHandle);
    client.DeleteVariableHandle(wStringHandle);
    }
}
```

5.3 Read/Write PlcOpen types (DATE, TIME ...)

The following PLCOpen types have specific representations within the TwinCAT.Ads Communication Library: DT [▶ 1570]

- DT [▶ 1570]
- DATE [▶ 1552]
- TIME [▶ 1598]
- LTIME [▶ 1581]
- TOD [▶ 1615]

The following section shows the different scenarios as code snippets.

HowTo Read/Write PlcOpen values

Reading writing by streams:

Read/Write PlcOpen types (streamed, async)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    ResultHandle resultHandleTime = await client.CreateVariableHandleAsync("MAIN.time", cancel); //
TIME
    ResultHandle resultHandleLTime = await client.CreateVariableHandleAsync("MAIN.lTime", cancel); /
/ LTIME
    ResultHandle resultHandleDate = await client.CreateVariableHandleAsync("MAIN.date", cancel); //
DATE

    if (resultHandleTime.Succeeded && resultHandleLTime.Succeeded && resultHandleDate.Succeeded)
    {
    try
    {
        byte[] readBuffer = new byte[LTIME.MarshalSize]; // Largest PlcOpen Type is 8 Bytes
        byte[] writeBuffer = new byte[LTIME.MarshalSize];

        // Reading raw value TIME
        await client.ReadAsync(resultHandleTime.Handle, readBuffer.AsMemory(0, TIME.MarshalSize), ca
ncel);

        // Unmarshalling
        TIME plcTime = null;
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, TIME.MarshalSize), out plcTime
);
        TimeSpan time = plcTime.Time;

        // Writing raw value TIME
        PrimitiveTypeMarshaler.Default.Marshal(time, writeBuffer.AsSpan());
        await client.WriteAsync(resultHandleTime.Handle, writeBuffer.AsMemory(0, TIME.MarshalSize),
cancel);

        // Reading raw value LTIME
        await client.ReadAsync(resultHandleLTime.Handle, readBuffer.AsMemory(0, LTIME.MarshalSize),
cancel);

        // Unmarshalling
        LTIME plcLTime = null;
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, LTIME.MarshalSize), out plcLTi
me);
        TimeSpan lTime = plcLTime.Time;

        // Writing raw value LTIME
        PrimitiveTypeMarshaler.Default.Marshal(lTime, writeBuffer.AsSpan());
        await client.WriteAsync(resultHandleLTime.Handle, writeBuffer.AsMemory(0, LTIME.MarshalSize)
, cancel);

        // Reading raw value DATE
        DATE plcDate = null;
        await client.ReadAsync(resultHandleDate.Handle, readBuffer.AsMemory(0, DATE.MarshalSize), ca
ncel);

        // Unmarshalling
        PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, DATE.MarshalSize), out plcDate
);
        DateTimeOffset dateTime = plcDate.Date;

        // Writeing raw value DATE
        PrimitiveTypeMarshaler.Default.Marshal(plcDate, writeBuffer.AsSpan());
        await client.WriteAsync(resultHandleDate.Handle, writeBuffer.AsMemory(0, DATE.MarshalSize),
cancel);
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleLTime.Handle, cancel);
        ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleTime.Handle, cancel);
        ResultAds r3 = await client.DeleteVariableHandleAsync(resultHandleDate.Handle, cancel);
    }
    }
}
```

Read/Write PlcOpen types (streamed, sync)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    uint handleTime = 0;
    uint handleLTime = 0;
    uint handleDate = 0;

    try
    {
    handleTime = client.CreateVariableHandle("MAIN.time"); // TIME
    handleLTime = client.CreateVariableHandle("MAIN.lTime"); // LTIME
    handleDate = client.CreateVariableHandle("MAIN.date"); // DATE

    byte[] readBuffer = new byte[LTIME.MarshalSize]; // Largest PlcOpen Type is 8 Bytes
    byte[] writeBuffer = new byte[LTIME.MarshalSize];

    // Reading raw value TIME
    client.Read(handleTime, readBuffer.AsMemory(0,TIME.MarshalSize));

    // Unmarshalling
    TIME plcTime = null;
    PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, TIME.MarshalSize), out plcTime);
    TimeSpan time = plcTime.Time;

    // Writing raw value TIME
    PrimitiveTypeMarshaler.Default.Marshal(time, writeBuffer.AsSpan());
    client.Write(handleTime, writeBuffer.AsMemory(0,TIME.MarshalSize));

    // Reading raw value LTIME
    client.Read(handleLTime, readBuffer.AsMemory(0, LTIME.MarshalSize));

    // Unmarshalling
    LTIME plcLTime = null;
    PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, LTIME.MarshalSize), out plcLTime);
    TimeSpan lTime = plcLTime.Time;

    // Writing raw value LTIME
    PrimitiveTypeMarshaler.Default.Marshal(lTime, writeBuffer.AsSpan());
    client.Write(handleLTime, writeBuffer.AsMemory(0, LTIME.MarshalSize));

    // Reading raw value DATE
    DATE plcDate = null;
    client.Read(handleDate, readBuffer.AsMemory(0, DATE.MarshalSize));

    // Unmarshalling
    PrimitiveTypeMarshaler.Default.Unmarshal(readBuffer.AsSpan(0, DATE.MarshalSize), out plcDate);
    DateTimeOffset dateTime = plcDate.Date;

    // Writeing raw value DATE
    PrimitiveTypeMarshaler.Default.Marshal(plcDate, writeBuffer.AsSpan());
    client.Write(handleDate, writeBuffer.AsMemory(0, DATE.MarshalSize));
    }
    finally
    {
    client.DeleteVariableHandle(handleLTime);
    client.DeleteVariableHandle(handleTime);
    client.DeleteVariableHandle(handleDate);
    }
}
```

Reading writing by ANY type concept:

Read/Write PlcOpen types (ANY, async)

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    ResultHandle resultHandleTime = await client.CreateVariableHandleAsync("MAIN.time",cancel); // T
IME
    ResultHandle resultHandleDate = await client.CreateVariableHandleAsync("MAIN.date", cancel); //
DATE
    ResultHandle resultHandleLTime = await client.CreateVariableHandleAsync("MAIN.lTime", cancel); /
/ LTIME
```

```
    if (resultHandleTime.Succeeded && resultHandleDate.Succeeded && resultHandleLTime.Succeeded)
    {
    try
    {
        ResultAnyValue resultTime = await client.ReadAnyAsync(resultHandleTime.Handle, typeof(TIME),
 cancel); // TIME

        TIME time = (TIME)resultTime.Value;
        TimeSpan timeSpan = time.Time;

        await client.WriteAnyAsync(resultHandleTime.Handle, time, cancel);

        ResultAnyValue resultData = await client.ReadAnyAsync(resultHandleDate.Handle, typeof(DATE),
 cancel); // DATE
        DATE date = (DATE)resultTime.Value;
        DateTimeOffset dateTime = date.Date;

        await client.WriteAnyAsync(resultHandleDate.Handle, date, cancel);

        ResultAnyValue resultLTime = await client.ReadAnyAsync(resultHandleLTime.Handle, typeof(LTIM
E), cancel); // LTIME
        LTIME lTime = (LTIME)resultTime.Value;
        TimeSpan lTimeSpan = lTime.Time;

        await client.WriteAnyAsync(resultHandleLTime.Handle, lTime, cancel);
    }
    finally
    {
        ResultAds r1 = await client.DeleteVariableHandleAsync(resultHandleTime.Handle,cancel);
        ResultAds r2 = await client.DeleteVariableHandleAsync(resultHandleDate.Handle,cancel);
        ResultAds r3 = await client.DeleteVariableHandleAsync(resultHandleLTime.Handle,cancel);
    }
    }
}
```

Read/Write PlcOpen types (ANY, sync)

```
using (AdsClient client = new AdsClient())
{
    client.Connect(AmsNetId.Local, 851); // Connect to local plc

    uint handleTime = 0;
    uint handleDate = 0;
    uint handleLTime = 0;

    try
    {
    handleTime = client.CreateVariableHandle("MAIN.time"); // TIME
    handleDate = client.CreateVariableHandle("MAIN.date"); // DATE
    handleLTime = client.CreateVariableHandle("MAIN.lTime"); // LTIME

    TIME time = (TIME)client.ReadAny(handleTime, typeof(TIME)); // TIME
    TimeSpan timeSpan = time.Time;
    client.WriteAny(handleTime, time);

    DATE date = (DATE)client.ReadAny(handleDate, typeof(DATE)); // DATE
    DateTimeOffset dateTime = date.Date;
    client.WriteAny(handleDate, date);

    LTIME ltime = (LTIME)client.ReadAny(handleLTime, typeof(LTIME)); // LTIME
    TimeSpan lTimeSpan = ltime.Time;
    client.WriteAny(handleLTime, ltime);
    }
    finally
    {
    client.DeleteVariableHandle(handleTime);
    client.DeleteVariableHandle(handleDate);
    client.DeleteVariableHandle(handleLTime);
    }
}
```

5.4 Event driven read with ADS Notifications

Use of ADS Notifications (Async)

Trigger on changed values by ADS Notifications

```
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification2;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];
    int size = sizeof(UInt32);

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new Notific
ationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Use of ADS Notifications (Synchronous)

Trigger on changed values by ADS Notifications

```
private void RegisterNotifications()
{
    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    try
    {
        // Notification to a DINT Type (UINT32)
        // Check for change every 200 ms

        int size = sizeof(UInt32);
        //byte[] notificationBuffer = new byte[sizeof(UInt32)];

        notificationHandle = client.AddDeviceNotification("MAIN.nCounter", size, new NotificationSet
```

```
tings(AdsTransMode.OnChange, 200, 0), null);
        Thread.Sleep(5000); // Sleep the main thread to get some (asynchronous Notifications)
    }
    finally
    {
        // Unregister the Event / Handle
        client.DeleteDeviceNotification(notificationHandle);
        client.AdsNotification -= Client_AdsNotification;
    }
    }
}

private void Client_AdsNotification(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

5.5 Reactive Read/Write with Reactive Extensions

Observation of Notifications

Notifications (address specified by InstancePath) will be received cyclically as defined in Default [▶ 984] and put into the Observer pipeline for further processing. This example takes 20 Notification samples before returning.

Observe for Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    }
    );

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Observation of Symbolic Notifications

This example determines a symbol via Symbolloader and samples its values by Notifications with customized NotificationSettings [▶ 979]. Again 20 samples are taken before the Observation finishes.

Observe for Symbol Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
```

```
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Observation of dynamic Symbol Notifications

Here, the symbol is determined via Symbolloader again, but now the Notifications will be processed as 'dynamic' values.

Observer for dynamic Symbol Notifications

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = (IDynamicSymbolLoader)SymbolLoaderFactory.Create(client, SymbolLoaderSettings
.DefaultDynamic);
    dynamic symbols = symbolLoader.SymbolsDynamic;
    dynamic cycleCount = symbols.TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    // Value objects can be dynamically (on the fly) created objects here (e.g. structs)
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    // We have to give the 'hint' about IValueSymbol here, that the CLR finds the Extension Method '
WhenValueChanged' during runtime.
    IDisposable subscription = ((IValueSymbol)cycleCount).WhenValueChanged().Take(20).Subscribe(valu
eObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Polling observer

A polling observer doesn't use ADS notifications, but instead the value read is triggered by a time interval (polling) or a customized trigger function (on request).

Polling observer

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));
```

```
    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Writing values with observable subject

In this example, a symbolic value is written in a static time interval (1 second). The writing stops after 10 values. The WriteValues [▶ 1116] extension method can be used to seamlessly bind value writing into a reactive application.

Writing values with observable subject

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

5.6 Upgrading existing ADS Application code (Version 4.X --> 5.X)

Code migration path for application code Beckhoff.TwinCAT.Ads 4.X --> 5.X

The Beckhoff.TwinCAT.Ads package Version 4.X implements the Beckhoff ADS API for the Microsoft .NET FullFramework (>= 4.0) / CompactFramework (>= 2.0). Newer .NET framework implementations are basing on '.NET Core' (e.g. .NET 5.0).

To support these newer flavours like .NET Core, .NET Standard together with its new features, a new Version 5.X of the Beckhoff.TwinCAT.Ads API is available.

Because new platform features need some breaking changes within the API for seamless support, the decision was taken to remove some legacy features and burdens in the reimplementation. Nevertheless most of the interfaces remained stable and only a few approaches changed.

The breaking changes and some guidlines will be discussed here.

Use 'async' wherever possible.

Internally, the Beckhoff.TwinCAT.Ads package are implemented asynchronously (Methodnames ending with 'Async'). The synchronous counterparts of the async methods are implemented as thin wrapper around the async method with synchronization.

As consequence the synchronous method is less flexible (only usable in synchronous code paths) and less performant (blocking calls, more thread switches). Furthermore, with the 'async' keyword support of .NET they are also not easier to use. So take more than a thought about migrate to async code.

Synchronous Read value

```
uint variableHandle = ...;
int value;
AdsErrorCode errorCode = adsClient.TryReadValue<int>(variableHandle, out int value);
```

Asynchronous

```
uint variableHandle = ...;
ResultValue<int> result = await adsClient.ReadValueAsync(variableHandle, CancellationToken.None);
AdsErrorCode errorCode = result.ErrorCode;
int value = result.Value;
```

However in synchronous application code paths the await keyword is not allowed. Using the synchronous methods is the only option besides migrating the whole application code up to the main root. Further information can be found here: Asynchronous programming with async and await

Name change TcAdsClient --> AdsClient

This should simply document that there is another implemenation. Just rename the class instance.

Version 4.X

```
TcAdsClient client = new TcAdsClient()
```

Version 5.X

```
AdsClient client = new AdsClient()
```

Marshalling of byte[] and missing AdsStream class.

Newer .NET implementations have more efficient possiblities to exchange memory data between software layers and classes (see Memory- and span-related types).

All API interface methods that use the combination of parameters (bytes, offset, length) or AdsStream

Version 4.X

```
public int TcAdsClient.Read(uint indexGroup,uint indexOffset,AdsStream dataStream)
```

Version 4.X

```
public int TcAdsClient.Write(uint indexGroup,uint indexOffset,AdsStream dataStream)
```

or

Version 4.X

```
public int TcAdsClient.Read(uint indexGroup,uint indexOffset,byte[] bytes, int offset, length)
```

Version 4.X

```
public int TcAdsClient.Write(uint indexGroup,uint indexOffset,byte[] bytes, int offset, length)
```

are changed like this:

Version 5.X

```
public int AdsClient.Read(uint indexGroup,uint indexOffset,Memory<byte> buffer)
```

Version 5.X

```
public int AdsClient.Write(uint indexGroup,uint indexOffset,ReadOnlyMemory<byte> buffer)
```

With this change a lot of interfaces will look much cleaner and act more efficient because of less array copy operations.

Handles are defined as uint type.

Handles are now streamlined consistently as uint type instead of int.

Version 4.X

```
public int TcAdsClient.CreateVariableHandle(string symbolPath)
```

Version 5.X

```
public uint AdsClient.CreateVariableHandle(string symbolPath)
```

Notification parameters bundled in one setings class.

Version 4.X

```
public int AddDeviceNotification(string symbolPath, AdsStream dataStream, int offset, int length, Ad
sTransMode transMode, int cycleTime, int maxDelay,Object userData)
```

Version 4.X

```
public int TcAdsClient.AddDeviceNotification(string symbolPath, AdsStream dataStream, int offset, in
t length, AdsTransMode transMode, int cycleTime, int maxDelay, Object userData)
```

The new parameterset simply needs the (byte) size of the transfered data instead of AdsStream, offset and length. Furthermore transMode, cycleTime and maxDelay are bundled in the NotificationSettings object.

Version 5.X

```
public uint AdsClient.AddDeviceNotification(string symbolPath,int dataSize, NotificationSettings set
tings, object userData)
```

6 TwinCAT.Ads Namespaces

TwinCAT ADS .NET API for .NET CORE

Namespaces

Namespace	Description
TwinCAT [▶ 55]	Common namespace for types that are not specific to ADS.
TwinCAT.Ads [▶ 151]	ADS root namespace.
TwinCAT.Ads.Reactive [▶ 1056]	Reactive Extensions for the ADS Client. All types within are contained in the ADS companion package "Beckhoff.TwinCAT.Ads.Reactive" which must be referenced seperately. https://www.nuget.org/packages/Beckhoff.TwinCAT.Ads.Reactive/
TwinCAT.Ads.Server [▶ 1121]	Namespace for the TwinCAT ADS Server Component.
TwinCAT.Ads.SumCommand [▶ 1205]	ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. ADS Sum-Commands offer to read/write with one single ADS call multiple variables which are not structured within a linear memory, effectively reducing roundtrips.
TwinCAT.Ads.TcpRouter [▶ 1254]	
TwinCAT.Ads.TypeSystem [▶ 1328]	Root namespace for the ADS type system.
TwinCAT.Ads.ValueAccess [▶ 1542]	Root namespace for ADS value access.
TwinCAT.Ams [▶ 1543]	
TwinCAT.PlcOpen [▶ 1552]	
TwinCAT.TypeSystem [▶ 1622]	Namespace for the common (non ADS dependent) type system.
TwinCAT.TypeSystem.Generic [▶ 2441]	Namespace for the dynamic part of the common type system.
TwinCAT.ValueAccess [▶ 2540]	Namespace for the common (non ADS dependent) value access.

6.1 TwinCAT Namespace

Common namespace for types that are not specific to ADS.

Classes

	Class	Description
	AdsException [▶ 57]	Base class for all exceptions thrown by the TwinCAT.Ads component
	ClientNotConnectedException [▶ 63]	Class ClientNotConnectedException
	ConnectionStateChangedEventArgs [▶ 68]	Event arguments for the Connection status changed event.

	Class	Description
	Session [▶ 101]	Abstract Session base class.
	SessionConnectionStateChangedEventArgs [▶ 116]	EventArguments for the ConnectionStatusChanged events.
	SessionException [▶ 122]	Session Exception
	SessionNotConnectedException [▶ 129]	Class SessionNotConnectedException.
	SessionProvider.TSession, TAddress, TSettings. [▶ 134]	Abstract base class for a Custom Session provider
	SymbolLoaderSettings [▶ 140]	Settings object for the SymbolLoader initialization.

Interfaces

	Interface	Description
	IConnection [▶ 74]	Interface IConnection
	IConnectionStateObserver [▶ 80]	Interface for a Connection state watcher (observer)
	IConnectionStateProvider [▶ 84]	Interface IConnectionStateProvider
	ISession [▶ 88]	Interface ISession
	ISessionProvider [▶ 94]	Interface ISessionProvider
	ISessionProvider.TSession, TAddress, TSettings. [▶ 97]	Generic ISessionProvider interface
	ISymbolLoaderSettings [▶ 99]	Interface ISymbolLoaderSettings
	ISymbolServerProvider [▶ 99]	Interface ISymbolServerProvider

Enumerations

	Enumeration	Description
	ConnectionState [▶ 67]	Connection state enumeration

	Enumeration	Description
	ConnectionStateChangedReason [▶ 73]	Reason for the Connection status changed event.
	SessionProviderCapabilities [▶ 140]	Enum SessionProviderCapabilities
	SymbolsLoadMode [▶ 149]	Enum SymbolsLoadMode
	ValueUpdateMode [▶ 150]	Value Update Mode.

6.1.1 AdsException Class

Base class for all exceptions thrown by the TwinCAT.Ads component

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException
      More... [▶ 59]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class AdsException : Exception
```

The AdsException type exposes the following members.

Constructors

	Name	Description
	AdsException. [▶ 59]	Initializes a new Instance of the AdsException class.
	AdsException(String) [▶ 60]	Initializes a new Instance of the AdsException class.
	AdsException(SerializationInfo, StreamingContext) [▶ 60]	Initializes a new instance of the AdsException class.
	AdsException(String, Exception) [▶ 61]	Initializes a new Instance of the AdsException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT Namespace [▶ 55]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException
      TwinCAT.Ads.AdsErrorException [▶ 583]
      TwinCAT.Ads.AdsInvalidNotificationException [▶ 591]
      TwinCAT.ClientNotConnectedException [▶ 63]
      TwinCAT.SessionException [▶ 122]
      TwinCAT.TypeSystem.DataTypeException [▶ 1658]
      TwinCAT.TypeSystem.MarshalException [▶ 2273]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]

6.1.1.1 AdsException Constructor

Overload List

	Name	Description
	AdsException. [▶ 59]	Initializes a new Instance of the AdsException [▶ 57] class.
	AdsException(String) [▶ 60]	Initializes a new Instance of the AdsException [▶ 57] class.
	AdsException(SerializationInfo, StreamingContext) [▶ 60]	Initializes a new instance of the AdsException [▶ 57] class.
	AdsException(String, Exception) [▶ 61]	Initializes a new Instance of the AdsException class.

Reference

AdsException Class [▶ 57]

TwinCAT Namespace [▶ 55]

6.1.1.1.1 AdsException Constructor

Initializes a new Instance of the AdsException [▶ 57] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsException()
```

Reference

AdsException Class [▶ 57]

AdsException Overload [▶ 59]

TwinCAT Namespace [▶ 55]

6.1.1.1.2 AdsException Constructor (String)

Initializes a new Instance of the AdsException [▶ 57] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsException(
    string message
)
```

Parameters

message	Type: System.String A message that describes the error.

Reference

AdsException Class [▶ 57]

AdsException Overload [▶ 59]

TwinCAT Namespace [▶ 55]

6.1.1.1.3 AdsException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the AdsException [▶ 57] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

AdsException Class [▶ 57]

AdsException Overload [▶ 59]

TwinCAT Namespace [▶ 55]

6.1.1.1.4 AdsException Constructor (String, Exception)

Initializes a new Instance of the AdsException class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsException(
    string message,
    Exception innerException
)
```

Parameters

message	Type: System.String The error message that explains the reason for the exception.
innerException	Type: System.Exception The exception that is the cause of the current exception. If the innerException parameter is not a null reference, the current exception is raised in a catch block that handles the inner exception.

Reference

AdsException Class [▶ 57]

AdsException Overload [▶ 59]

TwinCAT Namespace [▶ 55]

6.1.1.2 AdsException Properties

The AdsException [▶ 57] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

AdsException Class [▶ 57]

TwinCAT Namespace [▶ 55]

6.1.1.3 AdsException Methods

The AdsException [▶ 57] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AdsException Class [▶ 57]

TwinCAT Namespace [▶ 55]

6.1.1.4 AdsException Events

The AdsException [▶ 57] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

AdsException Class [▶ 57]

TwinCAT Namespace [▶ 55]

6.1.2 ClientNotConnectedException Class

Class ClientNotConnectedException

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.ClientNotConnectedException

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class ClientNotConnectedException : AdsException
```

The ClientNotConnectedException type exposes the following members.

Constructors

	Name	Description
	ClientNotConnectedException. [▶ 65]	Initializes a new instance of the ClientNotConnectedException class.
	ClientNotConnectedException(SerializationInfo, StreamingContext) [▶ 65]	Initializes a new instance of the ClientNotConnectedException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)

	Name	Description
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT Namespace [▶ 55]

6.1.2.1 ClientNotConnectedException Constructor

Overload List

	Name	Description
	ClientNotConnectedException. [▶ 65]	Initializes a new instance of the ClientNotConnectedException [▶ 63] class.

	Name	Description
	ClientNotConnectedException(SerializationInfo, StreamingContext) [▶ 65]	Initializes a new instance of the ClientNotConnectedException [▶ 63] class.

Reference

ClientNotConnectedException Class [▶ 63]

TwinCAT Namespace [▶ 55]

6.1.2.1.1 ClientNotConnectedException Constructor

Initializes a new instance of the ClientNotConnectedException [▶ 63] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ClientNotConnectedException()
```

Reference

ClientNotConnectedException Class [▶ 63]

ClientNotConnectedException Overload [▶ 64]

TwinCAT Namespace [▶ 55]

6.1.2.1.2 ClientNotConnectedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the ClientNotConnectedException [▶ 63] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected ClientNotConnectedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.

streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

ClientNotConnectedException Class [▶ 63]

ClientNotConnectedException Overload [▶ 64]

TwinCAT Namespace [▶ 55]

6.1.2.2 ClientNotConnectedException Properties

The ClientNotConnectedException [▶ 63] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

ClientNotConnectedException Class [▶ 63]

TwinCAT Namespace [▶ 55]

6.1.2.3 ClientNotConnectedException Methods

The ClientNotConnectedException [▶ 63] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

ClientNotConnectedException Class [▶ 63]

TwinCAT Namespace [▶ 55]

6.1.2.4 ClientNotConnectedException Events

The ClientNotConnectedException [▶ 63] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

ClientNotConnectedException Class [▶ 63]

TwinCAT Namespace [▶ 55]

6.1.3 ConnectionState Enumeration

Connection state enumeration

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum ConnectionState
```

Members

	Member name	Value	Description
	None	0	Unknown / Uninitialized
	Disconnected	1	Disconnected

	Member name	Value	Description
	Connected	2	Connected
	Lost	3	Connection lost

Reference

TwinCAT Namespace [▶ 55]

6.1.4 ConnectionStateChangedEventArgs Class

Event arguments for the Connection status changed event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.ConnectionStateChangedEventArgs
      TwinCAT.SessionConnectionStateChangedEventArgs [▶ 116]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ConnectionStateChangedEventArgs : EventArgs
```

The ConnectionStateChangedEventArgs type exposes the following members.

Constructors

	Name	Description
	ConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState) [▶ 70]	Constructs the ConnectionStateChangedEventArgs arguments.
	ConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, Exception) [▶ 70]	Constructs the ConnectionStateChangedEventArgs arguments.

Properties

	Name	Description
	Exception [▶ 71]	Exception, (only for Error [▶ 73]

	Name	Description
	NewState [▶ 72]	New connection state
	OldState [▶ 72]	Old connection state
	Reason [▶ 72]	Reason for the event

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT Namespace [▶ 55]

6.1.4.1 ConnectionStateChangedEventArgs Constructor

Overload List

	Name	Description
	ConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState) [▶ 70]	Constructs the ConnectionStateChangedEventArgs [▶ 68] arguments.
	ConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, Exception) [▶ 70]	Constructs the ConnectionStateChangedEventArgs [▶ 68] arguments.

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.1.1 ConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState)

Constructs the ConnectionStateChangedEventArgs [▶ 68] arguments.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState
)
```

Parameters

reason	Type: TwinCAT.ConnectionStateChangedReason [▶ 73] The reason.
newState	Type: TwinCAT.ConnectionState [▶ 67] The new state.
oldState	Type: TwinCAT.ConnectionState [▶ 67] The old state.

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

ConnectionStateChangedEventArgs Overload [▶ 69]

TwinCAT Namespace [▶ 55]

6.1.4.1.2 ConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState, Exception)

Constructs the ConnectionStateChangedEventArgs [▶ 68] arguments.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState,
    Exception e
)
```

Parameters

reason	Type: TwinCAT.ConnectionStateChangedReason [▶ 73] The reason.

newState	Type: TwinCAT.ConnectionState [▶ 67] The new state.
oldState	Type: TwinCAT.ConnectionState [▶ 67] The old state.
e	Type: System.Exception The e.

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

ConnectionStateChangedEventArgs Overload [▶ 69]

TwinCAT Namespace [▶ 55]

6.1.4.2 ConnectionStateChangedEventArgs Properties

The ConnectionStateChangedEventArgs [▶ 68] type exposes the following members.

Properties

	Name	Description
	Exception [▶ 71]	Exception, (only for Error [▶ 73]
	NewState [▶ 72]	New connection state
	OldState [▶ 72]	Old connection state
	Reason [▶ 72]	Reason for the event

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.2.1 ConnectionStateChangedEventArgs.Exception Property

Exception, (only for Error [▶ 73]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Exception Exception { get; }
```

Property Value

Type: Exception

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.2.2 ConnectionStateChangedEventArgs.NewState Property

New connection state

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionState NewState { get; }
```

Property Value

Type: ConnectionState [▶ 67]

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.2.3 ConnectionStateChangedEventArgs.OldState Property

Old connection state

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionState OldState { get; }
```

Property Value

Type: ConnectionState [▶ 67]

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.2.4 ConnectionStateChangedEventArgs.Reason Property

Reason for the event

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionStateChangedReason Reason { get; }
```

Property Value

Type: ConnectionStateChangedReason [▶ 73]

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.4.3 ConnectionStateChangedEventArgs Methods

The ConnectionStateChangedEventArgs [▶ 68] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ConnectionStateChangedEventArgs Class [▶ 68]

TwinCAT Namespace [▶ 55]

6.1.5 ConnectionStateChangedReason Enumeration

Reason for the Connection status changed event.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum ConnectionStateChangedReason
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Established	1	The connection to the target has been established
	Closed	2	The Connection was closed
	Lost	3	The connection to the target has been lost
	Error	4	Communication error to the target (the connection is not shutting down)
	Resurrected	5	Communication was resurrected (available again)

Reference

TwinCAT Namespace [▶ 55]

6.1.6 IConnection Interface

Interface IConnection

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IConnection : IConnectionStateProvider
```

The IConnection type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding.
	Id [▶ 76]	Gets the Connection Identifier .
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	Session [▶ 77]	Gets the session that initiated this IConnection

	Name	Description
	Timeout [▶ 77]	Gets the timeout (in milliseconds)

Methods

	Name	Description
	Close [▶ 78]	Closes this IConnection
	Connect [▶ 78]	(Re)Connects the IConnection when disconnected.
	Disconnect [▶ 79]	Disconnects this IConnection.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

TwinCAT Namespace [▶ 55]

6.1.6.1 IConnection Properties

The IConnection [▶ 74] type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding.
	Id [▶ 76]	Gets the Connection Identifier .
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74]
	Timeout [▶ 77]	Gets the timeout (in milliseconds)

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.1.1 IConnection.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.1.2 IConnection.Id Property

Gets the Connection Identifier .

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Id { get; }
```

Property Value

Type: Int32
The identifier.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.1.3 IConnection.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsConnected { get; }
```

Property Value

Type: Boolean

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.1.4 IConnection.Session Property

Gets the session that initiated this IConnection [▶ 74]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]
The session or NULL

Remarks

The Session can be null on standalone connections.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.1.5 IConnection.Timeout Property

Gets the timeout (in milliseconds)

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Timeout { get; set; }
```

Property Value

Type: Int32
The timeout.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.2 IConnection Methods

The IConnection [▶ 74] type exposes the following members.

Methods

	Name	Description
	Close [▶ 78]	Closes this IConnection [▶ 74]
	Connect [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected.
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74].

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.2.1 IConnection.Close Method

Closes this IConnection [▶ 74]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Close()
```

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.2.2 IConnection.Connect Method

(Re)Connects the IConnection [▶ 74] when disconnected.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Connect()
```

Return Value

Type: Boolean
true if connected, false otherwise.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.2.3 IConnection.Disconnect Method

Disconnects this IConnection [▶ 74].

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Disconnect()
```

Return Value

Type: Boolean
true if disconnected, false otherwise.

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.6.3 IConnection Events

The IConnection [▶ 74] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

IConnection Interface [▶ 74]

TwinCAT Namespace [▶ 55]

6.1.7 IConnectionStateObserver Interface

Interface for a Connection state watcher (observer)

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IConnectionStateObserver : IConnectionStateProvider
```

The IConnectionStateObserver type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	ErrorsSinceLastSucceeded [▶ 81]	Gets the number of errors since the last successful access
	LastAccess [▶ 81]	Gets the DateTimeOffset of the last tried access
	LastSucceededAccess [▶ 82]	Gets the DateTimeOffset of the last successful data communication
	Quality [▶ 82]	Gets the quality of the current cached value (the age of the data)
	TotalCycles [▶ 83]	Gets the number of successful reads / writes
	TotalErrors [▶ 83]	Gets the error count of accesses

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

TwinCAT Namespace [▶ 55]

6.1.7.1 IConnectionStateObserver Properties

The IConnectionStateObserver [▶ 80] type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	ErrorsSinceLastSucceeded [▶ 81]	Gets the number of errors since the last successful access
	LastAccess [▶ 81]	Gets the DateTimeOffset of the last tried access
	LastSucceededAccess [▶ 82]	Gets the DateTimeOffset of the last successful data communication
	Quality [▶ 82]	Gets the quality of the current cached value (the age of the data)
	TotalCycles [▶ 83]	Gets the number of successful reads / writes
	TotalErrors [▶ 83]	Gets the error count of accesses

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.1 IConnectionStateObserver.ErrorsSinceLastSucceeded Property

Gets the number of errors since the last successful access

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ErrorsSinceLastSucceeded { get; }
```

Property Value

Type: Int32

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.2 IConnectionStateObserver.LastAccess Property

Gets the DateTimeOffset of the last tried access

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DateTimeOffset LastAccess { get; }
```

Property Value

Type: DateTimeOffset

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.3 IConnectionStateObserver.LastSucceededAccess Property

Gets the DateTimeOffset of the last successful data communication

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DateTimeOffset LastSucceededAccess { get; }
```

Property Value

Type: DateTimeOffset

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.4 IConnectionStateObserver.Quality Property

Gets the quality of the current cached value (the age of the data)

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TimeSpan Quality { get; }
```

Property Value

Type: TimeSpan

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.5 IConnectionStateObserver.TotalCycles Property

Gets the number of successful reads / writes

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TotalCycles { get; }
```

Property Value

Type: Int32

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.1.6 IConnectionStateObserver.TotalErrors Property

Gets the error count of accesses

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TotalErrors { get; }
```

Property Value

Type: Int32

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.7.2 IConnectionStateObserver Events

The IConnectionStateObserver [▶ 80] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

IConnectionStateObserver Interface [▶ 80]

TwinCAT Namespace [▶ 55]

6.1.8 IConnectionStateProvider Interface

Interface IConnectionStateProvider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IConnectionStateProvider
```

The IConnectionStateProvider type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider has been changed.

Reference

TwinCAT Namespace [▶ 55]

6.1.8.1 IConnectionStateProvider Properties

The IConnectionStateProvider [▶ 84] type exposes the following members.

Properties

	Name	Description
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84]

Reference

IConnectionStateProvider Interface [▶ 84]

TwinCAT Namespace [▶ 55]

6.1.8.1.1 IConnectionStateProvider.ConnectionState Property

Gets the current Connection state of the IConnectionStateProvider [▶ 84]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ConnectionState ConnectionState { get; }
```

Property Value

Type: ConnectionState [▶ 67]
The state of the connection.

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
```

```
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

IConnectionStateProvider Interface [▶ 84]

TwinCAT Namespace [▶ 55]

IConnectionStateProvider.ConnectionStateChanged [▶ 86]

6.1.8.2 IConnectionStateProvider Events

The IConnectionStateProvider [▶ 84] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed.

Reference

IConnectionStateProvider Interface [▶ 84]

TwinCAT Namespace [▶ 55]

6.1.8.2.1 IConnectionStateProvider.ConnectionStateChanged Event

Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: System.EventHandler.ConnectionStateChangedEventArgs [▶ 68].

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState [▶ 85] updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

IConnectionStateProvider Interface [▶ 84]

TwinCAT Namespace [▶ 55]

IConnectionStateProvider.ConnectionState [▶ 85]

6.1.9 ISession Interface

Interface ISession

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISession : IConnectionStateProvider
```

The ISession type exposes the following members.

Properties

	Name	Description
	AddressSpecifier [▶ 89]	Gets the communication endpoint address string representation.
	Connection [▶ 90]	Gets the Connection object.
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	EstablishedAt [▶ 90]	Gets the UTC time when the session was established.
	Id [▶ 91]	Gets the Session Id
	IsConnected [▶ 91]	Gets a value indicating whether the session is connected.
	Provider [▶ 91]	Gets the Session Provider

Methods

	Name	Description
	Close [▶ 92]	Closes this ISession
	Connect [▶ 93]	Connects the session and returns the established IConnection [▶ 74] object.
	Disconnect [▶ 93]	Disconnects the ISession

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

TwinCAT Namespace [▶ 55]

6.1.9.1 ISession Properties

The ISession [▶ 88] type exposes the following members.

Properties

	Name	Description
	AddressSpecifier [▶ 89]	Gets the communication endpoint address string representation.
	Connection [▶ 90]	Gets the Connection object.
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	EstablishedAt [▶ 90]	Gets the UTC time when the session was established.
	Id [▶ 91]	Gets the Session Id
	IsConnected [▶ 91]	Gets a value indicating whether the session is connected.
	Provider [▶ 91]	Gets the Session Provider

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.1 ISession.AddressSpecifier Property

Gets the communication endpoint address string representation.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string AddressSpecifier { get; }
```

Property Value

Type: String
The address.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.2 ISession.Connection Property

Gets the Connection object.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]
The connection.

Remarks

The IConnection [▶ 74] object is established by the ISession [▶ 88] via Connect. [▶ 93] and is valid until the Disconnect. [▶ 93] method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently under the hood of the IConnection [▶ 74] so that the IConnection [▶ 74] instance and ISession [▶ 88] instance.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.3 ISession.EstablishedAt Property

Gets the UTC time when the session was established.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DateTimeOffset EstablishedAt { get; }
```

Property Value

Type: DateTimeOffset
The session established at.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.4 ISession.Id Property

Gets the Session Id

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Id { get; }
```

Property Value

Type: Int32
The identifier.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.5 ISession.IsConnected Property

Gets a value indicating whether the session is connected.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsConnected { get; }
```

Property Value

Type: Boolean
true if session is connected; otherwise, false.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.1.6 ISession.Provider Property

Gets the Session Provider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISessionProvider Provider { get; }
```

Property Value

Type: ISessionProvider [▶ 94]
The provider.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.2 ISession Methods

The ISession [▶ 88] type exposes the following members.

Methods

	Name	Description
	Close [▶ 92]	Closes this ISession [▶ 88]
	Connect [▶ 93]	Connects the session and returns the established IConnection [▶ 74] object.
	Disconnect [▶ 93]	Disconnects the ISession [▶ 88]

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.2.1 ISession.Close Method

Closes this ISession [▶ 88]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Close()
```

Remarks

Closes also the IConnection [▶ 74].

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.2.2 ISession.Connect Method

Connects the session and returns the established IConnection [▶ 74] object.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IConnection Connect()
```

Return Value

Type: IConnection [▶ 74]
The IConnection [▶ 74] object.

Remarks

The IConnection [▶ 74] will be valid until the ISession [▶ 88] is disconnected via the Disconnect. [▶ 93] method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently within the IConnection [▶ 74] so that the IConnection [▶ 74] instance and ISession [▶ 88] instance remains.

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.2.3 ISession.Disconnect Method

Disconnects the ISession [▶ 88]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Disconnect()
```

Return Value

Type: Boolean
true if Session was disconnected, false if the session was already closed.

Remarks

Disposes also the IConnection [▶ 74].

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.9.3 ISession Events

The ISession [▶ 88] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

ISession Interface [▶ 88]

TwinCAT Namespace [▶ 55]

6.1.10 ISessionProvider Interface

Interface ISessionProvider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISessionProvider
```

The ISessionProvider type exposes the following members.

Properties

	Name	Description
	Capabilities [▶ 95]	Gets the capabilities.
	Name [▶ 95]	Gets the name of the SessionProvider

Methods

	Name	Description
	Create [▶ 96]	Creates the Session with address and communication settings.

Remarks

Interface for Supporting Session / Communication providers

Reference

TwinCAT Namespace [▶ 55]

6.1.10.1 ISessionProvider Properties

The ISessionProvider [▶ 94] type exposes the following members.

Properties

	Name	Description
	Capabilities [▶ 95]	Gets the capabilities.
	Name [▶ 95]	Gets the name of the SessionProvider

Reference

ISessionProvider Interface [▶ 94]

TwinCAT Namespace [▶ 55]

6.1.10.1.1 ISessionProvider.Capabilities Property

Gets the capabilities.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
SessionProviderCapabilities Capabilities { get; }
```

Property Value

Type: SessionProviderCapabilities [▶ 140]
The capabilities.

Reference

ISessionProvider Interface [▶ 94]

TwinCAT Namespace [▶ 55]

6.1.10.1.2 ISessionProvider.Name Property

Gets the name of the SessionProvider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

ISessionProvider Interface [▶ 94]

TwinCAT Namespace [▶ 55]

6.1.10.2 ISessionProvider Methods

The ISessionProvider [▶ 94] type exposes the following members.

Methods

	Name	Description
	Create [▶ 96]	Creates the Session with address and communication settings.

Reference

ISessionProvider Interface [▶ 94]

TwinCAT Namespace [▶ 55]

6.1.10.2.1 ISessionProvider.Create Method

Creates the Session with address and communication settings.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISession Create(
    Object address,
    ISessionSettings settings
)
```

Parameters

address	Type: System.Object The address.
settings	Type: ISessionSettings The settings.

Return Value

Type: ISession [▶ 88]
ISession.

Reference

ISessionProvider Interface [▶ 94]

TwinCAT Namespace [▶ 55]

6.1.11 ISessionProvider.TSession, TAddress, TSettings. Interface

Generic ISessionProvider interface

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISessionProvider<TSession, TAddress, TSettings> : ISessionProvider
where TSession : ISession
```

Type Parameters

TSession	Session type
TAddress	Address specifier
TSettings	Communication settings type

The ISessionProvider.TSession, TAddress, TSettings. type exposes the following members.

Properties

	Name	Description
	Capabilities [▶ 95]	Gets the capabilities. (Inherited from ISessionProvider [▶ 94].)
	Name [▶ 95]	Gets the name of the SessionProvider (Inherited from ISessionProvider [▶ 94].)

Methods

	Name	Description
	Create(Object, ISessionSettings) [▶ 96]	Creates the Session with address and communication settings. (Inherited from ISessionProvider [▶ 94].)
	Create(TAddress, TSettings) [▶ 99]	Creates the Session with specified address and communication settings.

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.ISessionProvider [▶ 94]

6.1.11.1 ISessionProvider.TSession, TAddress, TSettings. Properties

The ISessionProvider.TSession, TAddress, TSettings. [▶ 97] generic type exposes the following members.

Properties

	Name	Description
	Capabilities [▶ 95]	Gets the capabilities. (Inherited from ISessionProvider [▶ 94].)
	Name [▶ 95]	Gets the name of the SessionProvider (Inherited from ISessionProvider [▶ 94].)

Reference

ISessionProvider.TSession, TAddress, TSettings. Interface [▶ 97]

TwinCAT Namespace [▶ 55]

6.1.11.2 ISessionProvider.TSession, TAddress, TSettings. Methods

Methods

	Name	Description
	Create(Object, ISessionSettings) [▶ 96]	Creates the Session with address and communication settings. (Inherited from ISessionProvider [▶ 94].)
	Create(TAddress, TSettings) [▶ 99]	Creates the Session with specified address and communication settings.

Reference

ISessionProvider.TSession, TAddress, TSettings. Interface [▶ 97]

TwinCAT Namespace [▶ 55]

6.1.11.2.1 ISessionProvider.TSession, TAddress, TSettings..Create Method

Overload List

	Name	Description
	Create(Object, ISessionSettings) [▶ 96]	Creates the Session with address and communication settings. (Inherited from ISessionProvider [▶ 94].)
	Create(TAddress, TSettings) [▶ 99]	Creates the Session with specified address and communication settings.

Reference

ISessionProvider.TSession, TAddress, TSettings. Interface [▶ 97]

TwinCAT Namespace [▶ 55]

ISessionProvider.TSession, TAddress, TSettings..Create Method (TAddress, TSettings)

Creates the Session with specified address and communication settings.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TSession Create(
    TAddress address,
    TSettings settings
)
```

Parameters

address	Type: TAddress [▶ 97] The address.
settings	Type: TSettings [▶ 97] The communicationSettings.

Return Value

Type: TSession [▶ 97]
S.

Reference

ISessionProvider.TSession, TAddress, TSettings. Interface [▶ 97]

Create Overload [▶ 98]

TwinCAT Namespace [▶ 55]

6.1.12 ISymbolLoaderSettings Interface

Interface ISymbolLoaderSettings

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolLoaderSettings
```

Reference

TwinCAT Namespace [▶ 55]

6.1.13 ISymbolServerProvider Interface

Interface ISymbolServerProvider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolServerProvider
```

The ISymbolServerProvider type exposes the following members.

Properties

	Name	Description
	SymbolServer [▶ 100]	Gets the symbol server.

Reference

TwinCAT Namespace [▶ 55]

6.1.13.1 ISymbolServerProvider Properties

The ISymbolServerProvider [▶ 99] type exposes the following members.

Properties

	Name	Description
	SymbolServer [▶ 100]	Gets the symbol server.

Reference

ISymbolServerProvider Interface [▶ 99]

TwinCAT Namespace [▶ 55]

6.1.13.1.1 ISymbolServerProvider.SymbolServer Property

Gets the symbol server.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolServer SymbolServer { get; }
```

Property Value

Type: ISymbolServer [▶ 2205]
The symbol server. To initially create this information the connection must be established.

Reference

ISymbolServerProvider Interface [▶ 99]

TwinCAT Namespace [▶ 55]

6.1.14 Session Class

Abstract Session base class.

Inheritance Hierarchy

System.Object
  TwinCAT.Session
    TwinCAT.Ads.AdsSessionBase [▶ 614]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class Session : ISession,
    IConnectionStateProvider, ISymbolServerProvider, IDisposable
```

The Session type exposes the following members.

Properties

	Name	Description
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation.
	Connection [▶ 104]	Gets the (established) connection.
	ConnectionState [▶ 104]	Gets the current Connection state of the Session
	Disposed [▶ 105]	Gets a value indicating whether this Session is disposed.
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established.
	Id [▶ 106]	Gets the Session Identifier
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected.
	Name [▶ 107]	Gets the name of the session
	SymbolServer [▶ 108]	Gets the symbol server.

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88]

	Name	Description
	Connect [▶ 110]	Connects the session.
	Disconnect [▶ 111]	Disconnects the session from the target.
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 112]	Releases unmanaged and - optionally - managed resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 112]	Gets the name/string identifier of the session.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 113]	Handler function connecting the Session.
	OnCreateSymbolServer [▶ 113]	Handler function creating the ISymbolServer [▶ 2205]
	OnDisconnect [▶ 114]	Handler function disconnecting the session.
	OnGetAddress [▶ 114]	Handler function getting the address of the session.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed.

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.ISession [▶ 88]

System.IDisposable

6.1.14.1 Session Properties

The Session [▶ 101] type exposes the following members.

Properties

	Name	Description
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation.
	Connection [▶ 104]	Gets the (established) connection.
	ConnectionState [▶ 104]	Gets the current Connection state of the Session [▶ 101]
	Disposed [▶ 105]	Gets a value indicating whether this Session [▶ 101] is disposed.
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established.
	Id [▶ 106]	Gets the Session Identifier
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected.
	Name [▶ 107]	Gets the name of the session
	SymbolServer [▶ 108]	Gets the symbol server.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.1 Session.AddressSpecifier Property

Gets the communication endpoint address string representation.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string AddressSpecifier { get; }
```

Property Value

Type: String
The address.

Implements

ISession.AddressSpecifier [▶ 89]

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.2 Session.Connection Property

Gets the (established) connection.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connection { get; protected set; }
```

Property Value

Type: IConnection [▶ 74]
The IConnection [▶ 74] if connection established, or **null** if not connected.

Implements

ISession.Connection [▶ 90]

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.3 Session.ConnectionState Property

Gets the current Connection state of the Session [▶ 101]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionState ConnectionState { get; }
```

Property Value

Type: ConnectionState [▶ 67]
The state of the connection.

Implements

IConnectionStateProvider.ConnectionState [▶ 85]

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

Session.ConnectionStateChanged [▶ 115]

6.1.14.1.4 Session.Disposed Property

Gets a value indicating whether this Session [▶ 101] is disposed.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Disposed { get; }
```

Property Value

Type: Boolean
true if disposed; otherwise, false.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.5 Session.EstablishedAt Property

Gets the UTC time when the session was established.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset EstablishedAt { get; }
```

Property Value

Type: DateTimeOffset
The session established at.

Implements

ISession.EstablishedAt [▶ 90]

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.6 Session.Id Property

Gets the Session Identifier

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Id { get; }
```

Property Value

Type: Int32
The identifier.

Implements

ISession.Id [▶ 91]

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.7 Session.IsConnected Property

Gets a value indicating whether this instance is connected.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

Type: Boolean
true if this instance is connected; otherwise, false.

Implements

ISession.IsConnected [▶ 91]

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.8 Session.Name Property

Gets the name of the session

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.1.9 Session.SymbolServer Property

Gets the symbol server.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolServer SymbolServer { get; }
```

Property Value

Type: ISymbolServer [▶ 2205]
The symbol server.

Implements

ISymbolServerProvider.SymbolServer [▶ 100]

Remarks

The Session [▶ 101] object holds and caches the symbolic information. To initially create this information, the Connection must be established.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2 Session Methods

The Session [▶ 101] type exposes the following members.

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88]
	Connect [▶ 110]	Connects the session.
	Disconnect [▶ 111]	Disconnects the session from the target.
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 112]	Releases unmanaged and - optionally - managed resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 112]	Gets the name/string identifier of the session.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 113]	Handler function connecting the Session.
	OnCreateSymbolServer [▶ 113]	Handler function creating the ISymbolServer [▶ 2205]
	OnDisconnect [▶ 114]	Handler function disconnecting the session.
	OnGetAddress [▶ 114]	Handler function getting the address of the session.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.1 Session.Close Method

Closes this ISession [▶ 88]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Close()
```

Implements

ISession.Close. [▶ 92]

Remarks

Closes also the IConnection [▶ 74].

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.2 Session.Connect Method

Connects the session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connect()
```

Return Value

Type: IConnection [▶ 74]
true if XXXX, false otherwise.

Implements

ISession.Connect. [▶ 93]

Exceptions

Exception	Condition
ObjectDisposedException	
ObjectDisposedException	

Remarks

The IConnection [▶ 74] will be valid until the ISession [▶ 88] is disconnected via the Disconnect. [▶ 111] method or the Dispose method is called. Any possible resurrections after communication losses will be done transparently within the IConnection [▶ 74] so that the IConnection [▶ 74] instance and ISession [▶ 88] instance remains.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.3 Session.Disconnect Method

Disconnects the session from the target.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Disconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

ISession.Disconnect. [▶ 93]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

Closes (and disposes) the underlying IConnection [▶ 74] The Session [▶ 101] itself will not be Disposed and can be reconnected.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.4 Session.Dispose Method

Overload List

	Name	Description
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 112]	Releases unmanaged and - optionally - managed resources.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

Session.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

Session Class [▶ 101]

Dispose Overload [▶ 111]

TwinCAT Namespace [▶ 55]

Session.Dispose Method (Boolean)

Releases unmanaged and - optionally - managed resources.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void Dispose(
    bool disposing
)
```

Parameters

disposing	Type: System.Boolean true to release both managed and unmanaged resources; false to release only unmanaged resources.

Reference

Session Class [▶ 101]

Dispose Overload [▶ 111]

TwinCAT Namespace [▶ 55]

6.1.14.2.5 Session.GetSessionName Method

Gets the name/string identifier of the session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected abstract string GetSessionName()
```

Return Value

Type: String
System.String.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.6 Session.OnConnect Method

Handler function connecting the Session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual IConnection OnConnect(
    bool reconnect
)
```

Parameters

reconnect	Type: System.Boolean if set to true [reconnect].

Return Value

Type: IConnection [▶ 74]
IConnection.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.7 Session.OnCreateSymbolServer Method

Handler function creating the ISymbolServer [▶ 2205]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected abstract ISymbolServer OnCreateSymbolServer()
```

Return Value

Type: ISymbolServer [▶ 2205]
ISymbolServer.

Exceptions

Exception	Condition
SessionNotConnectedException [▶ 129]	The connection is not established!

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.8 Session.OnDisconnect Method

Handler function disconnecting the session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual bool OnDisconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.2.9 Session.OnGetAddress Method

Handler function getting the address of the session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected abstract string OnGetAddress()
```

Return Value

Type: String
System.String.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.3 Session Events

The Session [▶ 101] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed.

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

6.1.14.3.1 Session.ConnectionStateChanged Event

Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: System.EventHandler.ConnectionStateChangedEventArgs [▶ 68].

Implements

IConnectionStateProvider.ConnectionStateChanged [▶ 86]

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState [▶ 104] updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

Session Class [▶ 101]

TwinCAT Namespace [▶ 55]

Session.ConnectionState [▶ 104]

6.1.15 SessionConnectionStateChangedEventArgs Class

EventArguments for the ConnectionStatusChanged events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.ConnectionStateChangedEventArgs [▶ 68]
      TwinCAT.SessionConnectionStateChangedEventArgs

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SessionConnectionStateChangedEventArgs : ConnectionStateChangedEventArgs
```

The SessionConnectionStateChangedEventArgs type exposes the following members.

Constructors

	Name	Description
	SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection) [▶ 119]	Initializes a new instance of the SessionConnectionStateChangedEventArgs class.
	SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception) [▶ 119]	Initializes a new instance of the SessionConnectionStateChangedEventArgs class.

Properties

	Name	Description
	Connection [▶ 121]	The connection
	Exception [▶ 71]	Exception, (only for Error [▶ 73] (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	NewState [▶ 72]	New connection state (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	OldState [▶ 72]	Old connection state (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	Reason [▶ 72]	Reason for the event (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	Session [▶ 121]	The session

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.ConnectionStateChangedEventArgs [▶ 68]

6.1.15.1 SessionConnectionStateChangedEventArgs Constructor

Overload List

	Name	Description
	SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection) [▶ 119]	Initializes a new instance of the SessionConnectionStateChangedEventArgs [▶ 116] class.
	SessionConnectionStateChangedEventArgs(ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception) [▶ 119]	Initializes a new instance of the SessionConnectionStateChangedEventArgs [▶ 116] class.

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

TwinCAT Namespace [▶ 55]

6.1.15.1.1 SessionConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection)

Initializes a new instance of the SessionConnectionStateChangedEventArgs [▶ 116] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
    ConnectionState oldState,
    ISession session,
    IConnection connection
)
```

Parameters

reason	Type: TwinCAT.ConnectionStateChangedReason [▶ 73] The reason.
newState	Type: TwinCAT.ConnectionState [▶ 67] The new state.
oldState	Type: TwinCAT.ConnectionState [▶ 67] The old state.
session	Type: TwinCAT.ISession [▶ 88] The session.
connection	Type: TwinCAT.IConnection [▶ 74] The connection.

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

SessionConnectionStateChangedEventArgs Overload [▶ 118]

TwinCAT Namespace [▶ 55]

6.1.15.1.2 SessionConnectionStateChangedEventArgs Constructor (ConnectionStateChangedReason, ConnectionState, ConnectionState, ISession, IConnection, Exception)

Initializes a new instance of the SessionConnectionStateChangedEventArgs [▶ 116] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionConnectionStateChangedEventArgs(
    ConnectionStateChangedReason reason,
    ConnectionState newState,
```

```
    ConnectionState oldState,
    ISession session,
    IConnection connection,
    Exception e
)
```

Parameters

reason	Type: TwinCAT.ConnectionStateChangedReason [▶ 73] The reason.
newState	Type: TwinCAT.ConnectionState [▶ 67] The new state.
oldState	Type: TwinCAT.ConnectionState [▶ 67] The old state.
session	Type: TwinCAT.ISession [▶ 88] The session.
connection	Type: TwinCAT.IConnection [▶ 74] The connection.
e	Type: System.Exception The e.

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

SessionConnectionStateChangedEventArgs Overload [▶ 118]

TwinCAT Namespace [▶ 55]

6.1.15.2 SessionConnectionStateChangedEventArgs Properties

The SessionConnectionStateChangedEventArgs [▶ 116] type exposes the following members.

Properties

	Name	Description
	Connection [▶ 121]	The connection
	Exception [▶ 71]	Exception, (only for Error [▶ 73] (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	NewState [▶ 72]	New connection state (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	OldState [▶ 72]	Old connection state (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	Reason [▶ 72]	Reason for the event (Inherited from ConnectionStateChangedEventArgs [▶ 68].)
	Session [▶ 121]	The session

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

TwinCAT Namespace [▶ 55]

6.1.15.2.1 SessionConnectionStateChangedEventArgs.Connection Property

The connection

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

TwinCAT Namespace [▶ 55]

6.1.15.2.2 SessionConnectionStateChangedEventArgs.Session Property

The session

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

TwinCAT Namespace [▶ 55]

6.1.15.3 SessionConnectionStateChangedEventArgs Methods

The SessionConnectionStateChangedEventArgs [▶ 116] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SessionConnectionStateChangedEventArgs Class [▶ 116]

TwinCAT Namespace [▶ 55]

6.1.16 SessionException Class

Session Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.SessionException
        TwinCAT.SessionNotConnectedException [▶ 129]

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class SessionException : AdsException
```

The SessionException type exposes the following members.

Constructors

	Name	Description
	SessionException(SerializationInfo, StreamingContext) [▶ 124]	Initializes a new instance of the SessionException class.
	SessionException(String, ISession) [▶ 125]	Initializes a new instance of the SessionException class.

	Name	Description
	SessionException(String, ISession, Exception) [▶ 125]	Initializes a new instance of the SessionException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Session [▶ 127]	Gets the session.
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 128]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.AdsException [▶ 57]

6.1.16.1 SessionException Constructor

Overload List

	Name	Description
	SessionException(SerializationInfo, StreamingContext) [▶ 124]	Initializes a new instance of the SessionException [▶ 122] class.
	SessionException(String, ISession) [▶ 125]	Initializes a new instance of the SessionException [▶ 122] class.
	SessionException(String, ISession, Exception) [▶ 125]	Initializes a new instance of the SessionException [▶ 122] class.

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.16.1.1 SessionException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SessionException [▶ 122] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected SessionException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

SessionException Class [▶ 122]

SessionException Overload [▶ 124]

TwinCAT Namespace [▶ 55]

6.1.16.1.2 SessionException Constructor (String, ISession)

Initializes a new instance of the SessionException [▶ 122] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionException(
    string message,
    ISession session
)
```

Parameters

message	Type: System.String The message.
session	Type: TwinCAT.ISession [▶ 88] The session.

Reference

SessionException Class [▶ 122]

SessionException Overload [▶ 124]

TwinCAT Namespace [▶ 55]

6.1.16.1.3 SessionException Constructor (String, ISession, Exception)

Initializes a new instance of the SessionException [▶ 122] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionException(
    string message,
    ISession session,
    Exception innerException
)
```

Parameters

message	Type: System.String The message.
session	Type: TwinCAT.ISession [▶ 88] The session.
innerException	Type: System.Exception The inner exception.

Reference

SessionException Class [▶ 122]

SessionException Overload [▶ 124]

TwinCAT Namespace [▶ 55]

6.1.16.2 SessionException Properties

The SessionException [▶ 122] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Session [▶ 127]	Gets the session.
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.16.2.1 SessionException.Session Property

Gets the session.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]
The session.

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.16.3 SessionException Methods

The SessionException [▶ 122] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 128]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.16.3.1 SessionException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.16.4 SessionException Events

The SessionException [▶ 122] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

SessionException Class [▶ 122]

TwinCAT Namespace [▶ 55]

6.1.17 SessionNotConnectedException Class

Class SessionNotConnectedException.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.SessionException [▶ 122]
        TwinCAT.SessionNotConnectedException

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class SessionNotConnectedException : SessionException
```

The SessionNotConnectedException type exposes the following members.

Constructors

	Name	Description
	SessionNotConnectedException(ISession) [▶ 131]	Initializes a new instance of the SessionNotConnectedException class.
	SessionNotConnectedException(SerializationInfo, StreamingContext) [▶ 132]	Initializes a new instance of the SessionNotConnectedException class.
	SessionNotConnectedException(String, ISession) [▶ 132]	Initializes a new instance of the SessionNotConnectedException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)

	Name	Description
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Session [▶ 127]	Gets the session. (Inherited from SessionException [▶ 122].)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 128]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SessionException [▶ 122].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.SessionException [▶ 122]

6.1.17.1 SessionNotConnectedException Constructor

Overload List

	Name	Description
	SessionNotConnectedException(ISession) [▶ 131]	Initializes a new instance of the SessionNotConnectedException [▶ 129] class.
	SessionNotConnectedException(SerializationInfo, StreamingContext) [▶ 132]	Initializes a new instance of the SessionNotConnectedException [▶ 129] class.
	SessionNotConnectedException(String, ISession) [▶ 132]	Initializes a new instance of the SessionNotConnectedException [▶ 129] class.

Reference

SessionNotConnectedException Class [▶ 129]

TwinCAT Namespace [▶ 55]

6.1.17.1.1 SessionNotConnectedException Constructor (ISession)

Initializes a new instance of the SessionNotConnectedException [▶ 129] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionNotConnectedException(
    ISession session
)
```

Parameters

session	Type: TwinCAT.ISession [▶ 88] The session.

Reference

SessionNotConnectedException Class [▶ 129]

SessionNotConnectedException Overload [▶ 131]

TwinCAT Namespace [▶ 55]

6.1.17.1.2 SessionNotConnectedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SessionNotConnectedException [▶ 129] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected SessionNotConnectedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

SessionNotConnectedException Class [▶ 129]

SessionNotConnectedException Overload [▶ 131]

TwinCAT Namespace [▶ 55]

6.1.17.1.3 SessionNotConnectedException Constructor (String, ISession)

Initializes a new instance of the SessionNotConnectedException [▶ 129] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionNotConnectedException(
    string message,
    ISession session
)
```

Parameters

message	Type: System.String The message.
session	Type: TwinCAT.ISession [▶ 88] The session.

Reference

SessionNotConnectedException Class [▶ 129]

SessionNotConnectedException Overload [▶ 131]

TwinCAT Namespace [▶ 55]

6.1.17.2 SessionNotConnectedException Properties

The SessionNotConnectedException [▶ 129] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Session [▶ 127]	Gets the session. (Inherited from SessionException [▶ 122].)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

SessionNotConnectedException Class [▶ 129]

TwinCAT Namespace [▶ 55]

6.1.17.3 SessionNotConnectedException Methods

The SessionNotConnectedException [▶ 129] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 128]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SessionException [▶ 122].)

	Name	Description
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

SessionNotConnectedException Class [▶ 129]

TwinCAT Namespace [▶ 55]

6.1.17.4 SessionNotConnectedException Events

The SessionNotConnectedException [▶ 129] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

SessionNotConnectedException Class [▶ 129]

TwinCAT Namespace [▶ 55]

6.1.18 SessionProvider.TSession, TAddress, TSettings. Class

Abstract base class for a Custom Session provider

Inheritance Hierarchy

System.Object
  TwinCAT.SessionProvider.TSession, TAddress, TSettings.
Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class SessionProvider<TSession, TAddress, TSettings> : ISessionProvider<TSession, TA
ddress, TSettings>,
    ISessionProvider
where TSession : ISession
where TSettings : class
```

Type Parameters

TSession	SessionType
TAddress	Address type
TSettings	Communication settings type

The SessionProvider.TSession, TAddress, TSettings. type exposes the following members.

Constructors

	Name	Description
	SessionProvider.TSession, TAddress, TSettings.. [▶ 136]	Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.
	SessionProvider.TSession, TAddress, TSettings. (SessionProviderCapabilities) [▶ 136]	Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. class.

Properties

	Name	Description
	Capabilities [▶ 137]	Gets the capabilities.
	Name [▶ 138]	Gets the name of the SessionProvider

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Fields

	Name	Description
S	s_self [▶ 139]	Singleton Instance.

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.ISessionProvider.TSession, TAddress, TSettings. [▶ 97]

6.1.18.1 SessionProvider.TSession, TAddress, TSettings. Constructor

Overload List

	Name	Description
	SessionProvider.TSession, TAddress, TSettings.. [▶ 136]	Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. [▶ 134] class.
	SessionProvider.TSession, TAddress, TSettings. (SessionProviderCapabilities) [▶ 136]	Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. [▶ 134] class.

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.1.1 SessionProvider.TSession, TAddress, TSettings. Constructor

Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. [▶ 134] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected SessionProvider()
```

Exceptions

Exception	Condition
Exception	Session provider already instantiated!

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

SessionProvider.TSession, TAddress, TSettings. Overload [▶ 136]

TwinCAT Namespace [▶ 55]

6.1.18.1.2 SessionProvider.TSession, TAddress, TSettings. Constructor (SessionProviderCapabilities)

Initializes a new instance of the SessionProvider.TSession, TAddress, TSettings. [▶ 134] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected SessionProvider(
    SessionProviderCapabilities cap
)
```

Parameters

cap	Type: TwinCAT.SessionProviderCapabilities [▶ 140]

Exceptions

Exception	Condition
Exception	Session provider already instantiated!

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

SessionProvider.TSession, TAddress, TSettings. Overload [▶ 136]

TwinCAT Namespace [▶ 55]

6.1.18.2 SessionProvider.TSession, TAddress, TSettings. Properties

The SessionProvider.TSession, TAddress, TSettings. [▶ 134] generic type exposes the following members.

Properties

	Name	Description
	Capabilities [▶ 137]	Gets the capabilities.
	Name [▶ 138]	Gets the name of the SessionProvider

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.2.1 SessionProvider.TSession, TAddress, TSettings..Capabilities Property

Gets the capabilities.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionProviderCapabilities Capabilities { get; }
```

Property Value

Type: SessionProviderCapabilities [▶ 140]
The capabilities.

Implements

ISessionProvider.Capabilities [▶ 95]

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.2.2 SessionProvider.TSession, TAddress, TSettings..Name Property

Gets the name of the SessionProvider

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract string Name { get; }
```

Property Value

Type: String
The name.

Implements

ISessionProvider.Name [▶ 95]

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.3 SessionProvider.TSession, TAddress, TSettings. Methods

The SessionProvider.TSession, TAddress, TSettings. [▶ 134] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.4 SessionProvider.TSession, TAddress, TSettings. Fields

The SessionProvider.TSession, TAddress, TSettings. [▶ 134] generic type exposes the following members.

Fields

	Name	Description
	s_self [▶ 139]	Singleton Instance.

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.18.4.1 SessionProvider.TSession, TAddress, TSettings..s_self Field

Singleton Instance.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected static ISessionProvider<TSession, TAddress, TSettings> s_self
```

Field Value

Type: ISessionProvider [▶ 97].TSession [▶ 134], TAddress [▶ 134], TSettings [▶ 134].

Reference

SessionProvider.TSession, TAddress, TSettings. Class [▶ 134]

TwinCAT Namespace [▶ 55]

6.1.19 SessionProviderCapabilities Enumeration

Enum SessionProviderCapabilities

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum SessionProviderCapabilities
```

Members

	Member name	Value	Description
	DataTypeSupport	1	Supports DataTypes
	SymbolBrowsing	2	Supports Symbol Browsing
	ValueRead	4	Supports VAlue Read
	ValueWrite	8	Support Value Write
	ValueNotifications	16	Supports Value changed Notifications
	None	0	Uninitilialized / None
	Mask_All	31	All Capabilities active

Reference

TwinCAT Namespace [▶ 55]

6.1.20 SymbolLoaderSettings Class

Settings object for the SymbolLoader initialization.

Inheritance Hierarchy

System.Object
  TwinCAT.SymbolLoaderSettings
Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SymbolLoaderSettings : ISymbolLoaderSettings
```

The SymbolLoaderSettings type exposes the following members.

Constructors

	Name	Description
	SymbolLoaderSettings(SymbolsLoadMode) [▶ 143]	Initializes a new instance of the SymbolLoaderSettings class with IndexGroupOffsetPreferred [▶ 1542].

	Name	Description
	SymbolLoaderSettings(SymbolsLoadMode, ValueAccessMode) [▶ 143]	Initializes a new instance of the SymbolLoaderSettings class.
	SymbolLoaderSettings(SymbolsLoadMode, ValueCreationModes, ValueAccessMode) [▶ 144]	Initializes a new instance of the SymbolLoaderSettings class.

Properties

	Name	Description
	AutomaticReconnection [▶ 145]	Gets or sets a value indicating whether Disconnect connections can be reconnected.
S	Default [▶ 145]	Gets the default settings object for standard symbols.
S	DefaultDynamic [▶ 146]	Gets the default settings object for Dynamic symbols.
	NonCachedArrayElements [▶ 147]	Gets or sets the setting to create ArrayElements "On-The-Fly" (Default True)
	SymbolsLoadMode [▶ 147]	Gets or sets the symbols load mode.
	ValueAccessMode [▶ 147]	Gets or sets the value access mode.
	ValueCreation [▶ 148]	Gets or sets the value creation mode.
	ValueUpdateMode [▶ 148]	Gets or sets the value update mode.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This settings object is used for the initialization of the internal Symbol loader object.

Reference

TwinCAT Namespace [▶ 55]

TwinCAT.ISymbolLoaderSettings [▶ 99]

TwinCAT.Ads.TypeSystem.SymbolLoaderFactory [▶ 1523]

TwinCAT.TypeSystem.ISymbolLoader [▶ 2200]

TwinCAT.SymbolsLoadMode [▶ 149]

TwinCAT.Ads.ValueAccess.ValueAccessMode [▶ 1542]

6.1.20.1 SymbolLoaderSettings Constructor

Overload List

	Name	Description
	SymbolLoaderSettings(SymbolsLoadMode) [▶ 143]	Initializes a new instance of the SymbolLoaderSettings [▶ 140] class with IndexGroupOffsetPreferred [▶ 1542].
	SymbolLoaderSettings(SymbolsLoadMode, ValueAccessMode) [▶ 143]	Initializes a new instance of the SymbolLoaderSettings [▶ 140] class.
	SymbolLoaderSettings(SymbolsLoadMode, ValueCreationModes, ValueAccessMode) [▶ 144]	Initializes a new instance of the SymbolLoaderSettings [▶ 140] class.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.1.1 SymbolLoaderSettings Constructor (SymbolsLoadMode)

Initializes a new instance of the SymbolLoaderSettings [▶ 140] class with IndexGroupOffsetPreferred [▶ 1542].

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode
)
```

Parameters

loadMode	Type: TwinCAT.SymbolsLoadMode [▶ 149] The load mode.

Reference

SymbolLoaderSettings Class [▶ 140]

SymbolLoaderSettings Overload [▶ 142]

TwinCAT Namespace [▶ 55]

6.1.20.1.2 SymbolLoaderSettings Constructor (SymbolsLoadMode, ValueAccessMode)

Initializes a new instance of the SymbolLoaderSettings [▶ 140] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode,
    ValueAccessMode valueAccess
)
```

Parameters

loadMode	Type: TwinCAT.SymbolsLoadMode [▶ 149] The load mode.
valueAccess	Type: TwinCAT.Ads.ValueAccess.ValueAccessMode [▶ 1542] The value access.

Reference

SymbolLoaderSettings Class [▶ 140]

SymbolLoaderSettings Overload [▶ 142]

TwinCAT Namespace [▶ 55]

6.1.20.1.3 SymbolLoaderSettings Constructor (SymbolsLoadMode, ValueCreationModes, ValueAccessMode)

Initializes a new instance of the SymbolLoaderSettings [▶ 140] class.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolLoaderSettings(
    SymbolsLoadMode loadMode,
    ValueCreationModes valueCreation,
    ValueAccessMode valueAccess
)
```

Parameters

loadMode	Type: TwinCAT.SymbolsLoadMode [▶ 149] The load mode.
valueCreation	Type: TwinCAT.ValueAccess.ValueCreationModes [▶ 2580] The dynamic value creation.
valueAccess	Type: TwinCAT.Ads.ValueAccess.ValueAccessMode [▶ 1542] The value access.

Reference

SymbolLoaderSettings Class [▶ 140]

SymbolLoaderSettings Overload [▶ 142]

TwinCAT Namespace [▶ 55]

6.1.20.2 SymbolLoaderSettings Properties

The SymbolLoaderSettings [▶ 140] type exposes the following members.

Properties

	Name	Description
	AutomaticReconnection [▶ 145]	Gets or sets a value indicating whether Disconnect connections can be reconnected.
S	Default [▶ 145]	Gets the default settings object for standard symbols.
S	DefaultDynamic [▶ 146]	Gets the default settings object for Dynamic symbols.
	NonCachedArrayElements [▶ 147]	Gets or sets the setting to create ArrayElements "On-The-Fly" (Default True)
	SymbolsLoadMode [▶ 147]	Gets or sets the symbols load mode.

	Name	Description
	ValueAccessMode [▶ 147]	Gets or sets the value access mode.
	ValueCreation [▶ 148]	Gets or sets the value creation mode.
	ValueUpdateMode [▶ 148]	Gets or sets the value update mode.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.1 SymbolLoaderSettings.AutomaticReconnection Property

Gets or sets a value indicating whether Disconnect connections can be reconnected.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool AutomaticReconnection { get; set; }
```

Property Value

Type: Boolean
true if Disconnect connections can be reconnecte; otherwise, false.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.2 SymbolLoaderSettings.Default Property

Gets the default settings object for standard symbols.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SymbolLoaderSettings Default { get; }
```

Property Value

Type: SymbolLoaderSettings [▶ 140]
The default settings object.

Remarks

The following defaults are set here:

Setting	Description
Symbols load mode (SymbolsLoadMode [▶ 147])	Create virtual tree (VirtualTree [▶ 149].
Value access mode (ValueAccessMode [▶ 147])	Prefer Symbolic access of values (Symbolic [▶ 1542]).
Value creation mode ValueCreation [▶ 148]	Create .NET integral primitives if possible (Default [▶ 2580].

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

SymbolLoaderSettings.DefaultDynamic [▶ 146]

6.1.20.2.3 SymbolLoaderSettings.DefaultDynamic Property

Gets the default settings object for Dynamic symbols.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SymbolLoaderSettings DefaultDynamic { get; }
```

Property Value

Type: SymbolLoaderSettings [▶ 140]
The dynamic default settings object.

Remarks

The following defaults are set here:

Setting	Description
Symbols load mode (SymbolsLoadMode [▶ 147])	Create dynamic tree (DynamicTree [▶ 149].
Value access mode (ValueAccessMode [▶ 147])	Prefer Symbolic access of values (Symbolic [▶ 1542]).
Value creation mode ValueCreation [▶ 148]	Create .NET integral primitives if possible (Default [▶ 2580].

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

SymbolLoaderSettings.Default [▶ 145]

6.1.20.2.4 SymbolLoaderSettings.NonCachedArrayElements Property

Gets or sets the setting to create ArrayElements "On-The-Fly" (Default True)

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool NonCachedArrayElements { get; set; }
```

Property Value

Type: Boolean
The value access mode.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.5 SymbolLoaderSettings.SymbolsLoadMode Property

Gets or sets the symbols load mode.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolsLoadMode SymbolsLoadMode { get; set; }
```

Property Value

Type: SymbolsLoadMode [▶ 149]
The symbols load mode.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.6 SymbolLoaderSettings.ValueAccessMode Property

Gets or sets the value access mode.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ValueAccessMode ValueAccessMode { get; set; }
```

Property Value

Type: ValueAccessMode [▶ 1542]
The value access mode.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.7 SymbolLoaderSettings.ValueCreation Property

Gets or sets the value creation mode.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ValueCreationModes ValueCreation { get; set; }
```

Property Value

Type: ValueCreationModes [▶ 2580]
The dynamic value mode.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.2.8 SymbolLoaderSettings.ValueUpdateMode Property

Gets or sets the value update mode.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ValueUpdateMode ValueUpdateMode { get; set; }
```

Property Value

Type: ValueUpdateMode [▶ 150]
The value update mode.

Remarks

Mode	Description
Immediately [▶ 150]	Writes the values of this DynamicValue [▶ 1869]"/> instantly when setting its value or the value of its child members/elements.
Triggered [▶ 150]	Caches internally the value of this DynamicValue [▶ 1869]"/> until the Write. [▶ 1893] method is called. This reduces ADS rountrips, if one or more member/ element values should be changed. Furthermore the write on the destination system happens consistently in one ADS Write operation, which could be important for dependent properties/members/elements.

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.20.3 SymbolLoaderSettings Methods

The SymbolLoaderSettings [▶ 140] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SymbolLoaderSettings Class [▶ 140]

TwinCAT Namespace [▶ 55]

6.1.21 SymbolsLoadMode Enumeration

Enum SymbolsLoadMode

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum SymbolsLoadMode
```

Members

	Member name	Value	Description
	Flat	0	Loads the Symbols organized as Flat List
	VirtualTree	1	Loads the Symbols organized as Virtual tree (Symbol Parent - Child relationships)
	DynamicTree	2	Loads the Symbols as a Virtual tree with Dynamic Symbols (Only available within versions > 4.X of this ADS Api)

Reference

TwinCAT Namespace [▶ 55]

6.1.22 ValueUpdateMode Enumeration

Value Update Mode.

Namespace: TwinCAT [▶ 55]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum ValueUpdateMode
```

Members

	Member name	Value	Description
	None	0	No automatic Value Update / Uninitialized
	Immediately	1	Update Value immediately on property set access.
	Triggered	2	Triggers the ValueUpdate explicitly

Remarks

Mode	Description
Immediately	Writes the values of this DynamicValue instantly when setting its value or the value of its child members/elements.
Triggered	Caches internally the value of this DynamicValue until the DynamicValue.Write method is called. This reduces ADS rountrips, if one or more member/element values should be changed. Furthermore the write on the destination system happens consistently in one ADS Write operation, which could be important for dependent properties/members/elements.

Reference

TwinCAT Namespace [▶ 55]

6.2 TwinCAT.Ads Namespace

ADS root namespace.

Classes

	Class	Description
	AdsClient [▶ 154]	ADS Client / ADS Communication object.
	AdsClientSettings [▶ 344]	Settings object for the AdsClient [▶ 154] class.
	AdsCommunicationStatistics [▶ 350]	ADS Communication statistics
	AdsConnection [▶ 357]	ADS Connection class
	AdsDataTypeArrayInfo [▶ 572]	Array definition for a single dimension.
	AdsErrorCodeExtensions [▶ 581]	Class AdsErrorCodeExtensions.
	AdsErrorException [▶ 583]	The exception that is thrown when an ADS error occurs.
	AdsInvalidNotificationException [▶ 591]	This AdsInvalidNotificationException is created if the length of the notification data is 0. This indicates that the notification handle is not valid any more. This exception is passed to the AdsNotificationErrorEvent.
	AdsNotificationErrorEventArgs [▶ 595]	Arguments for the AdsNotificationError [▶ 867] events.
	AdsNotificationEventArgs [▶ 597]	Event argument class for AdsNotification [▶ 866] events.
	AdsNotificationExEventArgs [▶ 601]	Arguments for AdsNotificationEx [▶ 868] events.
	AdsSession [▶ 603]	AdsSession class
	AdsSessionBase [▶ 614]	Abstract base class for ADS Sessions.
	AdsStateChangedEventArgs [▶ 627]	Arguments for the AdsStateChanged [▶ 934] event.

	Class	Description
	AdsStateChangedEventArgs2 [▶ 629]	Event Arguments for AdsStateChanged events.
	AdsSumCommandException [▶ 632]	The exception that is thrown when an ADS SumCommandBase error occurs.
	AdsSymbolVersionChangedEventArgs [▶ 637]	Arguments for the AdsSymbolVersionChanged [▶ 937] event.
	AdsVersion [▶ 642]	The structure contains the version number, revision number and build number.
	AmsAddress [▶ 648]	Ams/Ads Address
	AmsNetId [▶ 665]	AMS/ADS Net ID
	AmsRouterNotificationEventArgs [▶ 695]	Arguments for the IRouterNotificationProvider [▶ 973] events.
	DeviceInfo [▶ 698]	The structure contains the name and the version information of the device.
	Notification [▶ 974]	Class Notification. Implements the INotification [▶ 969]
	NotificationSettings [▶ 979]	Notification communication settings
	ResultAds [▶ 989]	Base class for an (asynchronous) ADS Task Result
	ResultAnyValue [▶ 998]	Result object for asynchronous reading an 'AnyValue'/Primitive Value via tasks.
	ResultDeviceInfo [▶ 1001]	Ads Task Result for DeviceInfo [▶ 1003] requests (async operations).
	ResultHandle [▶ 1005]	Result object for asynchronous registering an ADS Handle via tasks.
	ResultRead [▶ 1008]	Asynchronous ADS Read result.
	ResultReadAdsState [▶ 1010]	Result object for asynchronous reading AdsStates via tasks.
	ResultReadBytes [▶ 1013]	ADS Task Result returning Read data for async Read operations.
	ResultReadDeviceState [▶ 1016]	Result object for asynchronous ADS ReadDeviceState tasks.
	ResultReadWrite [▶ 1019]	Result object for asynchronous ADS ReadWrite tasks.
	ResultReadWriteBytes [▶ 1022]	Result object for asynchronous ADS ReadWrite tasks.
	ResultRpcMethod [▶ 1025]	Class representing a result of an asynchronous RpcMethod call. Implements the ResultAds [▶ 989]
	ResultValue.TValue. [▶ 1029]	ADS Result object returning a generic value result (TValue) (asynchronous read). Implements the ResultAds [▶ 989]

	Class	Description
	ResultWrite [▶ 1032]	Result for asynchronous ADS write tasks.
	SessionSettings [▶ 1035]	Session settings class
	TaskExtensions [▶ 1049]	TaskExtensions for Task Cancellation and Timeout
	ValueNotificationEventArgs.T. [▶ 1053]	Arguments for AdsNotificationEx [▶ 868] events.

Structures

	Structure	Description
	StateInfo [▶ 1041]	The structure contains the ADS state and device state.

Interfaces

	Interface	Description
	IAdsAnyAccess [▶ 702]	Interface for accessing ADS 'Any' objects.
	IAdsConnectAddress [▶ 733]	Interface for method to connect the ADS client via AMS Address.
	IAdsConnection [▶ 765]	ADS Connection interface
	IAdsDisposableConnection [▶ 797]	Interface IAdsDisposableConnection Implements the IAdsConnectAddress [▶ 733] Implements the IRouterNotificationProvider [▶ 973] Implements the IAdsSymbolChangedProvider [▶ 934] Implements the IDisposable
	IAdsHandle [▶ 827]	Interface for ADS access via variable handle
	IAdsNotifications [▶ 839]	Interface for Notification management.
	IAdsReadWrite [▶ 870]	Interface for ADS Read/Write access via IndexGroup / IndexOffset
	IAdsReadWrite2 [▶ 875]	Interface for ADS Read/Write access via IndexGroup / IndexOffset
	IAdsReadWriteTimeoutAccess [▶ 879]	Interface IAdsReadWriteTimeoutAccess
	IAdsRpcInvoke [▶ 886]	Interface IAdsRpcInvoke
	IAdsSession [▶ 908]	Interface IAdsSession
	IAdsSessionSettings [▶ 913]	Interface for ADS Session Settings
	IAdsStateControl [▶ 915]	Interface for reading and controlling the ADS state.
	IAdsStateControlTimeout [▶ 921]	Interface IAdsStateControlTimeout

	Interface	Description
	IAdsStateObserver [▶ 927]	Interface for an AdsState observer
	IAdsStateProvider [▶ 929]	Interface IAdsStateProvider
	IAdsSymbolChangedProvider [▶ 934]	Interface IAdsConnectionLegacy
	IAdsSymbolicAccess [▶ 937]	Interface for symbolic ads access.
	IAdsSymbolTableProvider [▶ 967]	Interface IAdsSymbolTableProvider
	INotification [▶ 969]	Common INotification interface
	INotificationSettings [▶ 972]	Interface for Notification Settings Implements the IComparable.T.
	IRouterNotificationProvider [▶ 973]	Interface for AMS Router Notifications.

Enumerations

	Enumeration	Description
	AdsCommandId [▶ 349]	AdsCommandId Enumeration
	AdsDataTypeId [▶ 574]	ADS data types.
	AdsErrorCode [▶ 575]	Describes the ADS error that occurred.
	AdsState [▶ 626]	Describes the AdsState.
	AdsTransMode [▶ 639]	ADS Transmission Mode for ADS Notifications.
	AmsPort [▶ 693]	AmsPorts
	AmsRouterState [▶ 697]	State of the AMS Router.
	TransportProtocols [▶ 1053]	Enum ADS TransportProtocol

6.2.1 AdsClient Class

ADS Client / ADS Communication object.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsClient
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsClient : IAdsDisposableConnection,
    IAdsConnectAddress, IAdsConnection, IConnection, IConnectionStateProvider, IAdsNotifications,
    IAdsSymbolicAccess, IAdsAnyAccess, IAdsHandle, IAdsReadWrite2, IAdsReadWrite,
    IAdsStateProvider, IAdsStateControl, IAdsSymbolChangedProvider, IAdsRpcInvoke, IRouterNotificati
onProvider,
    IDisposable, IAdsSymbolTableProvider
```

The AdsClient type exposes the following members.

Properties

	Name	Description
	Address [▶ 175]	Gets the target AmsAddress [▶ 648] of of the established ADS connection (Destination side).
	ClientAddress [▶ 176]	Get the client AmsAddress [▶ 648] (Source side).
	DefaultValueEncoding [▶ 176]	Gets the default value encoding.
	Id [▶ 177]	Gets the AdsClient Identifier.
	IsConnected [▶ 177]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	IsDisposed [▶ 178]	Gets a value indicating whether this instance is disposed.
	IsLocal [▶ 178]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.
	Session [▶ 179]	Gets the session that initiated this IConnection [▶ 74]
	SymbolEncoding [▶ 179]	Gets the symbol encoding.
	Timeout [▶ 180]	Sets the timeout for the ads communication. Unit is in ms.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 195]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 196]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 198]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 338] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 199]	Adds a device notification as an asynchronous operation.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 201]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 202]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 204]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 205]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 206]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

	Name	Description
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 208]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	CleanupSymbolTable [▶ 209]	Clears the internal symbol cache.
	Close [▶ 210]	Closes this AdsClient
	Connect(Int32) [▶ 211]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 211]	Connects the target
	Connect(AmsPort) [▶ 212]	Connects to the local target ADS Device.
	Connect(String, Int32) [▶ 212]	Connects to the target ADS Device.
	Connect(AmsNetId, Int32) [▶ 213]	Connects to the target ADS Device.
	Connect(AmsNetId, AmsPort) [▶ 214]	Connects to the target ADS Device.
	ConnectAndWaitAsync [▶ 214]	Connects to the target address and waits until the AdsClient is disconnected asynchronously.
	CreateVariableHandle [▶ 215]	Determines the Symbol handle by its instance path synchronously.
	CreateVariableHandleAsync [▶ 216]	Determines the Symbol handle by its instance path asynchronously.
	DeleteDeviceNotification [▶ 216]	Deletes a registered notification.
	DeleteDeviceNotificationAsync [▶ 217]	Deletes a registered notification asynchronously.
	DeleteVariableHandle [▶ 218]	Releases the specified symbol/variable handle synchronously.
	DeleteVariableHandleAsync [▶ 219]	Releases the specified symbol/variable handle asynchronously.
	Disconnect [▶ 220]	Disconnects this AdsClient from the local ADS router.
	Dispose [▶ 220]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 221]	Finalizes an instance of the AdsClient class. (Overrides Object.Finalize.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	InvokeRpcMethod(String, String, .Object.) [▶ 222]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 223]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 225]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 228]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 230]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 231]	invoke RPC method as an asynchronous operation.
	Read(UInt32, Memory) [▶ 233]	
	Read(UInt32, UInt32, Memory) [▶ 233]	

	Name	Description
	ReadAny(UInt32, Type) [▶ 236]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 238]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 240]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 241]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 237]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 239]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 245]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 247]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 243]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 244]	Reads data asynchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 246]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 247]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyString(UInt32, Int32, Encoding) [▶ 251]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 251]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 253]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 254]	Reads a string from a specified address asynchronously.
	ReadAsync(UInt32, Memory, Void) [▶ 255]	
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 256]	
	ReadDataType [▶ 256]	Call this method to obtain information about the specified data type.
	ReadDataTypeAsync [▶ 257]	read data type as an asynchronous operation.
	ReadDeviceInfo [▶ 258]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 259]	Reads the identification and version number of an ADS server.
	ReadState [▶ 260]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 260]	Reads the ADS status and the device status from an ADS server.
	ReadSymbol [▶ 261]	Call this method to obtain information about the individual symbols (variables) in ADS devices.

	Name	Description
	ReadSymbolAsync [▶ 261]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 263]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 265]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(String) [▶ 263]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 264]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 267]	Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync(String, Type, CancellationToken) [▶ 269]	Reads the value of a symbol asynchronously.
	ReadValueAsync.TValue.(String, CancellationToken) [▶ 266]	Reads the value of a symbol asynchronously.
	ReadValueAsync.TValue.(ISymbol, CancellationToken) [▶ 268]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadWrite(UInt32, Memory, Void) [▶ 270]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 271]	
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 272]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 273]	
	RegisterAdsStateChangedAsync [▶ 273]	Registers for AdsStateChanged [▶ 340] events as an asynchronous operation.

	Name	Description
	RegisterSymbolVersionChangedAsync [▶ 274]	Registers for an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 276]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 277]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 279]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 280]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryCreateVariableHandle [▶ 281]	Determines the Symbol handle by its instance path synchronously.
	TryDeleteDeviceNotification(UInt32) [▶ 283]	Deletes a registered notification.
	TryDeleteDeviceNotification(UInt32, Int32) [▶ 284]	Deletes a registered notification.
	TryDeleteVariableHandle [▶ 285]	Releases the specified symbol/variable handle synchronously.
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 286]	Invokes the specified RPC Method

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 288]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 290]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier.., AnyTypeSpecifier, .Object.., Object.) [▶ 291]	Tries the invoke RPC method.
	TryRead(UInt32, Memory, Void) [▶ 293]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 294]	
	TryReadDataType [▶ 294]	Call this method to obtain information about the specified data type.
	TryReadState [▶ 295]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadSymbol [▶ 296]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	TryReadValue(ISymbol, Object.) [▶ 298]	Reads the value of a symbol and returns it as an object.
	TryReadValue(String, Type, Object.) [▶ 300]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 297]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 299]	Reads the value of a symbol and returns it as an object.
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 301]	

	Name	Description
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 302]	
	TryResurrect [▶ 302]	Resurrects the connection
	TryWrite(UInt32, ReadOnlyMemory) [▶ 303]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 304]	
	TryWriteControl(StateInfo) [▶ 305]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 306]	
	TryWriteValue(String, Object) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 308]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 309]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	UnregisterAdsStateChangedAsync [▶ 310]	unregister ads state changed as an asynchronous operation.
	UnregisterSymbolVersionChangedAsync [▶ 311]	Unregisters from an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.
	Write(UInt32, ReadOnlyMemory) [▶ 312]	
	Write(UInt32, UInt32) [▶ 312]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 313]	

	Name	Description
	WriteAny(UInt32, Object) [▶ 314]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 315]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 315]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 316]	Writes an object synchronously to an ADS device.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 318]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 319]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 320]	Write the value of an Anytype (Primitive type) asynchronously.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 321]	Write the value of an Anytype (Primitive type) asynchronously.
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 322]	Writes the string (Potentially unsafe!)
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 323]	Writes the string (Potentially unsafe!)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 324]	

	Name	Description
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 325]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 326]	
	WriteControl(StateInfo) [▶ 327]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 328]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 328]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 329]	
	WriteSymbolAsync [▶ 330]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(String, Object) [▶ 331]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 332]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(String, T) [▶ 332]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue.T.(ISymbol, T) [▶ 333]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 335]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 334]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

	Name	Description
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 336]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Events

	Name	Description
	AdsNotification [▶ 338]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 339]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 340]	Occurs when the ADS devices sends a notification to the client.
	AdsStateChanged [▶ 340]	Occurs when the ADS state changes.
	AdsSumNotification [▶ 341]	Occurs when Notifications are send (bundled notifications)
	AdsSymbolVersionChanged [▶ 342]	Occurs when the symbol version has been changed changes.
	ConnectionStateChanged [▶ 343]	Occurs when the connection state has been changed.
	RouterStateChanged [▶ 344]	Occurs when the state of the local Router has changed.

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenSymbolVersionChanges(ISchedulers) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Remarks

The class AdsClient enables synchronous/asynchronous access to data of an ADS Device.

Examples

The following sample shows how to instantiate and use the AdsClient class.

AdsClient Demo (async)

```
using System;
using System.Buffers.Binary;
using System.Threading;
using System.Threading.Tasks;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;

namespace Sample
{
    class ClientAsync
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        AmsAddress address = ArgParser.Parse(args);
        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {

        // Connect to Address
        client.Connect(address.NetId,address.Port); // Connect to Port (851, first PLC by default)

        // Read the identification and version number of the device
        ResultDeviceInfo resultDeviceInfo = await client.ReadDeviceInfoAsync(cancel);

        if (resultDeviceInfo.Succeeded)
        {
            DeviceInfo deviceInfo = resultDeviceInfo.DeviceInfo;
            Version version = deviceInfo.Version.ConvertToStandard();
            Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
            Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));
        }
        // Read the state of the device
        ResultReadDeviceState resultReadDeviceState  = await client.ReadStateAsync(cancel);
```

```
        AdsState state = AdsState.Invalid;

        if (resultReadDeviceState.Succeeded)
        {
            StateInfo stateInfo = resultReadDeviceState.State;

            state = stateInfo.AdsState;
            short deviceState = stateInfo.DeviceState;
            Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
            Console.WriteLine(string.Format("AdsState   : {0}", state));
        }

        // Write ADS Commands (write state) to target
        // Set PLC to Run

        if (state == AdsState.Stop)
        {
            await client.WriteControlAsync(AdsState.Run,0,cancel);
        }

        //create variable handle for Plc Project Name (automatic generated symbol in PLC)
        ResultHandle resultHandle = await client.CreateVariableHandleAsync("TwinCAT_SystemInfoVarLis
t._AppInfo.ProjectName",cancel);

        if (resultHandle.Succeeded)
        {
            uint handleProjectName = resultHandle.Handle;
            uint handleNotification = 0; // Notification Handle for Task1 CycleCount changes

            try
            {
            // Read value from target and Marshal data into string
            byte[] readData = new byte[256];

                ResultRead resultRead = await client.ReadAsync(handleProjectName,readData.AsMemory()
, cancel);

                if (resultRead.Succeeded)
                {
                PrimitiveTypeMarshaler marshaler = PrimitiveTypeMarshaler.Default;

                string projectName = null;
                int unmarshaledBytes = marshaler.Unmarshal(readData, out projectName);
                Console.WriteLine(string.Format("ProjectName  : {0}", projectName));
                }

            //_notificationBuffer = new byte[4]; // Sizeof UDINT
            int size = sizeof(UInt32); // Sizeof UDINT

            client.AdsNotification += client_NotificationEvent; // Register for Notification event

            //Register Notification for Task1 CycleCount symbol (automatic generated symbol in PLC)
            resultHandle = await client.AddDeviceNotificationAsync("TwinCAT_SystemInfoVarList._TaskI
nfo[1].CycleCount", size, NotificationSettings.Default, null, cancel);

            if (resultHandle.Succeeded)
            {
                handleNotification = resultHandle.Handle;
                // Sleep 10 Seconds to receive events
                await Task.Delay(10000, cancel); // Async Delay (some time for notifications)
            }
            }
            finally
            {
            // Cleanup all handles
            // Dispose all Streams

            ResultAds result = await client.DeleteDeviceNotificationAsync(handleNotification, cancel
);

            // Always delete all variable handles.
            result = await client.DeleteVariableHandleAsync(handleProjectName, cancel);
            }
        }
        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
        }
    }
```

AdsClient Demo (sync)

```
using System;
using System.Buffers.Binary;
using System.Threading;
using System.Threading.Tasks;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;

namespace Sample
{
    class Client
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        AmsAddress address = ArgParser.Parse(args);

        using (AdsClient client = new AdsClient())
        {

        // Connect to Address
        client.Connect(address.NetId, address.Port); // Connect to Port (851, first PLC by default)

        // Read the identification and version number of the device
        DeviceInfo deviceInfo = client.ReadDeviceInfo();

        Version version = deviceInfo.Version.ConvertToStandard();
        Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
        Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));

        // Read the state of the device
        StateInfo stateInfo = client.ReadState();
        AdsState state = AdsState.Invalid;

        state = stateInfo.AdsState;
        short deviceState = stateInfo.DeviceState;
        Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
        Console.WriteLine(string.Format("AdsState   : {0}", state));

        // Write ADS Commands (write state) to target
        // Set PLC to Run

        if (state == AdsState.Stop)
        {
            client.WriteControl(new StateInfo(AdsState.Run, 0));
        }

        //create variable handle for Plc Project Name (automatic generated symbol in PLC)
        uint handleProjectName = client.CreateVariableHandle("TwinCAT_SystemInfoVarList._AppInfo.Pro
jectName");

        uint handleNotification = 0; // Notification Handle for Task1 CycleCount changes

        try
        {
            // Read value from target and Marshal data into string
            byte[] readData = new byte[256];
            int readBytes = client.Read(handleProjectName, readData.AsMemory());

            PrimitiveTypeMarshaler marshaler = PrimitiveTypeMarshaler.Default;

            string projectName = null;
            int unmarshaledBytes = marshaler.Unmarshal(readData, client.DefaultValueEncoding, out pr
ojectName);
            Console.WriteLine(string.Format("ProjectName  : {0}", projectName));

            //_notificationBuffer = new byte[4]; // Sizeof UDINT
            int size = sizeof(UInt32); // Sizeof UDINT

            client.AdsNotification += client_NotificationEvent; // Register for Notification event

            //Register Notification for Task1 CycleCount symbol (automatic generated symbol in PLC)
            handleNotification = client.AddDeviceNotification("TwinCAT_SystemInfoVarList._TaskInfo[1
].CycleCount", size, NotificationSettings.Default, null);

            // Sleep 10 Seconds to receive events
```

```
            Thread.Sleep(10000); // Sleep to get notifications
        }
        finally
        {
            // Cleanup all handles
            // Dispose all Streams
            client.DeleteDeviceNotification(handleNotification);

            // Always delete all variable handles.
            client.DeleteVariableHandle(handleProjectName);
        }
        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
        }
    }
```

Argument Parser

```
public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
    AmsNetId netId = AmsNetId.Local;
    int port = 851;

    if (args != null)
    {
        if (args.Length > 0 && args[0] != null)
        netId = AmsNetId.Parse(args[0]);

        if (args.Length > 1 && args[1] != null)
        port = int.Parse(args[1]);
    }
    return new AmsAddress(netId, port);
    }
}
```

The following sample shows how to call (Remote Procedures / Methods) within the PLC directly from the AdsClient class.

RPC Call Example (async)

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT.Ads;
    using TwinCAT.TypeSystem;

    class RpcCallAsync
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        //Parse the AmsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        // Create the ADS Client
        using (AdsClient client = new AdsClient())
        {
        // Establish Connection
        client.Connect(address);

        // Call a Method that has the following signature (within MAIN Program)

        /*  {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
```

```
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
        */

        ResultRpcMethod result1 = await client.InvokeRpcMethodAsync("MAIN", "M_Add", new object[] {
(short)1, (short)4 }, CancellationToken.None);
        short result = (short)result1.ReturnValue;

        // Call a Method that has no parameter and returns VOID
        await client.InvokeRpcMethodAsync("MAIN", "M_Method1", new object[] { },CancellationToken.No
ne);
        }
    }
    }
}
```

RPC Call Example (sync)

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using TwinCAT.Ads;
    using TwinCAT.TypeSystem;

    class RpcCall
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        //Parse the AmsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        // Create the ADS Client
        using (AdsClient client = new AdsClient())
        {
        // Establish Connection
        client.Connect(address);

        // Call a Method that has the following signature (within MAIN Program)

        /*  {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
        */

        short result = (short)client.InvokeRpcMethod("MAIN", "M_Add", new object[] { (short)1, (shor
t)4 });

        // Call a Method that has no parameter and returns VOID
        client.InvokeRpcMethod("MAIN", "M_Method1", new object[] { });
        }
    }
    }
}
```

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1 AdsClient Properties

The AdsClient [▶ 154] type exposes the following members.

Properties

	Name	Description
	Address [▶ 175]	Gets the target AmsAddress [▶ 648] of of the established ADS connection (Destination side).
	ClientAddress [▶ 176]	Get the client AmsAddress [▶ 648] (Source side).
	DefaultValueEncoding [▶ 176]	Gets the default value encoding.
	Id [▶ 177]	Gets the AdsClient [▶ 154] Identifier.
	IsConnected [▶ 177]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	IsDisposed [▶ 178]	Gets a value indicating whether this instance is disposed.
	IsLocal [▶ 178]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.
	Session [▶ 179]	Gets the session that initiated this IConnection [▶ 74]
	SymbolEncoding [▶ 179]	Gets the symbol encoding.
	Timeout [▶ 180]	Sets the timeout for the ads communication. Unit is in ms.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.1 AdsClient.Address Property

Gets the target AmsAddress [▶ 648] of of the established ADS connection (Destination side).

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The address.

Implements

IAdsConnection.Address [▶ 781]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.2 AdsClient.ClientAddress Property

Get the client AmsAddress [▶ 648] (Source side).

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress ClientAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The client address.

Implements

IAdsConnection.ClientAddress [▶ 781]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.3 AdsClient.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.

Implements

IConnection.DefaultValueEncoding [▶ 76]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.4 AdsClient.Id Property

Gets the AdsClient [▶ 154] Identifier.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Id { get; }
```

Property Value

Type: Int32
The identifier.

Implements

IConnection.Id [▶ 76]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.5 AdsClient.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

Type: Boolean
true if this instance is connected; otherwise, false.

Implements

IConnection.IsConnected [▶ 76]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.6 AdsClient.IsDisposed Property

Gets a value indicating whether this instance is disposed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

Type: Boolean
true if this instance is disposed; otherwise, false.

Implements

IAdsDisposableConnection.IsDisposed [▶ 812]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.7 AdsClient.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsLocal { get; }
```

Property Value

Type: Boolean
true if this instance is local; otherwise, false.

Implements

IAdsConnection.IsLocal [▶ 782]

Exceptions

Exception	Condition
NotImplementedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.8 AdsClient.Session Property

Gets the session that initiated this IConnection [▶ 74]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]
The session or NULL

Implements

IConnection.Session [▶ 77]

Remarks

The Session can be null on standalone connections.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.9 AdsClient.SymbolEncoding Property

Gets the symbol encoding.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding SymbolEncoding { get; }
```

Property Value

Type: Encoding
The symbol encoding.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.1.10 AdsClient.Timeout Property

Sets the timeout for the ads communication. Unit is in ms.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Timeout { get; set; }
```

Property Value

Type: Int32

Implements

IConnection.Timeout [▶ 77]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2 AdsClient Methods

The AdsClient [▶ 154] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 195]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.

	Name	Description
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 196]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 198]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 338] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 199]	Adds a device notification as an asynchronous operation.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 201]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 202]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 204]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 205]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

	Name	Description
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 206]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 208]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	CleanupSymbolTable [▶ 209]	Clears the internal symbol cache.
	Close [▶ 210]	Closes this AdsClient [▶ 154]
	Connect(Int32) [▶ 211]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 211]	Connects the target
	Connect(AmsPort) [▶ 212]	Connects to the local target ADS Device.
	Connect(String, Int32) [▶ 212]	Connects to the target ADS Device.
	Connect(AmsNetId, Int32) [▶ 213]	Connects to the target ADS Device.
	Connect(AmsNetId, AmsPort) [▶ 214]	Connects to the target ADS Device.
	ConnectAndWaitAsync [▶ 214]	Connects to the target address and waits until the AdsClient [▶ 154] is disconnected asynchronously.
	CreateVariableHandle [▶ 215]	Determines the Symbol handle by its instance path synchronously.
	CreateVariableHandleAsync [▶ 216]	Determines the Symbol handle by its instance path asynchronously.
	DeleteDeviceNotification [▶ 216]	Deletes a registered notification.
	DeleteDeviceNotificationAsync [▶ 217]	Deletes a registered notification asynchronously.
	DeleteVariableHandle [▶ 218]	Releases the specified symbol/variable handle synchronously.

	Name	Description
	DeleteVariableHandleAsync [▶ 219]	Releases the specified symbol/variable handle asynchronously.
	Disconnect [▶ 220]	Disconnects this AdsClient [▶ 154] from the local ADS router.
	Dispose [▶ 220]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 221]	Finalizes an instance of the AdsClient [▶ 154] class. (Overrides Object.Finalize..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	InvokeRpcMethod(String, String, .Object.) [▶ 222]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 223]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 225]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 228]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 230]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 231]	invoke RPC method as an asynchronous operation.
	Read(UInt32, Memory) [▶ 233]	
	Read(UInt32, UInt32, Memory) [▶ 233]	
	ReadAny(UInt32, Type) [▶ 236]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 238]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 240]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 241]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 237]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 239]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 245]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 247]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.

	Name	Description
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 243]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 244]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 246]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 247]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyString(UInt32, Int32, Encoding) [▶ 251]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 251]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 253]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 254]	Reads a string from a specified address asynchronously.
	ReadAsync(UInt32, Memory, Void) [▶ 255]	
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 256]	
	ReadDataType [▶ 256]	Call this method to obtain information about the specified data type.

	Name	Description
	ReadDataTypeAsync [▶ 257]	read data type as an asynchronous operation.
	ReadDeviceInfo [▶ 258]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 259]	Reads the identification and version number of an ADS server.
	ReadState [▶ 260]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 260]	Reads the ADS status and the device status from an ADS server.
	ReadSymbol [▶ 261]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadSymbolAsync [▶ 261]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 263]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 265]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(String) [▶ 263]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 264]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 267]	Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync(String, Type, CancellationToken) [▶ 269]	Reads the value of a symbol asynchronously.
	ReadValueAsync.TValue.(String, CancellationToken) [▶ 266]	Reads the value of a symbol asynchronously.
	ReadValueAsync.TValue.(ISymbol, CancellationToken) [▶ 268]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadWrite(UInt32, Memory, Void) [▶ 270]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 271]	

	Name	Description
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 272]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 273]	
	RegisterAdsStateChangedAsync [▶ 273]	Registers for AdsStateChanged [▶ 340] events as an asynchronous operation.
	RegisterSymbolVersionChangedAsync [▶ 274]	Registers for an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 276]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 277]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 279]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 280]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryCreateVariableHandle [▶ 281]	Determines the Symbol handle by its instance path synchronously.
	TryDeleteDeviceNotification(UInt32) [▶ 283]	Deletes a registered notification.

	Name	Description
	TryDeleteDeviceNotification(UInt32, Int32) [▶ 284]	Deletes a registered notification.
	TryDeleteVariableHandle [▶ 285]	Releases the specified symbol/variable handle synchronously.
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 286]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 288]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 290]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 291]	Tries the invoke RPC method.
	TryRead(UInt32, Memory, Void) [▶ 293]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 294]	
	TryReadDataType [▶ 294]	Call this method to obtain information about the specified data type.
	TryReadState [▶ 295]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadSymbol [▶ 296]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	TryReadValue(ISymbol, Object.) [▶ 298]	Reads the value of a symbol and returns it as an object.

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 300]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 297]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 299]	Reads the value of a symbol and returns it as an object.
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 301]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 302]	
	TryResurrect [▶ 302]	Resurrects the connection
	TryWrite(UInt32, ReadOnlyMemory) [▶ 303]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 304]	
	TryWriteControl(StateInfo) [▶ 305]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 306]	
	TryWriteValue(String, Object) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 308]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 309]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	UnregisterAdsStateChangedAsync [▶ 310]	unregister ads state changed as an asynchronous operation.

	Name	Description
	UnregisterSymbolVersionChangedAsync [▶ 311]	Unregisters from an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.
	Write(UInt32, ReadOnlyMemory) [▶ 312]	
	Write(UInt32, UInt32) [▶ 312]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 313]	
	WriteAny(UInt32, Object) [▶ 314]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 315]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 315]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 316]	Writes an object synchronously to an ADS device.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 318]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 319]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 320]	Write the value of an Anytype (Primitive type) asynchronously.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 321]	Write the value of an Anytype (Primitive type) asynchronously.

	Name	Description
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 322]	Writes the string (Potentially unsafe!)
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 323]	Writes the string (Potentially unsafe!)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 324]	
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 325]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 326]	
	WriteControl(StateInfo) [▶ 327]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 328]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 328]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 329]	
	WriteSymbolAsync [▶ 330]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(String, Object) [▶ 331]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 332]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(String, T) [▶ 332]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

	Name	Description
	WriteValue.T. (ISymbol, T) [▶ 333]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 335]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T. (String, T, CancellationToken) [▶ 334]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValueAsync.T. (ISymbol, T, CancellationToken) [▶ 336]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(ISchedulr) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.1 AdsClient.AddDeviceNotification Method

Overload List

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 195]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 196]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

symbolPath	Type: System.String Symbol / Instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.

userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotification Overload [▶ 195]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotification [▶ 338]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsClient.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotification Overload [▶ 195]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AdsClient.AdsNotification [▶ 338]

AdsClient.AdsNotificationError [▶ 339]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.1.2.2 AdsClient.AddDeviceNotificationAsync Method

Overload List

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 198]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 338] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 199]	Adds a device notification as an asynchronous operation.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AddDeviceNotificationAsync Method (String, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 338] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationAsync(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.

userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217] should always be called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationAsync Overload [▶ 198]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotification [▶ 338]

AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsClient.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)

Adds a device notification as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always be called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationAsync Overload [▶ 198]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

6.2.1.2.3 AdsClient.AddDeviceNotificationEx Method

Overload List

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 201]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 202]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 204]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 205]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationEx Overload [▶ 201]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsClient.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationEx Overload [▶ 201]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsClient.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationEx Overload [▶ 201]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.AdsNotificationError [▶ 339]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

AdsClient.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument.
args	Type: .System.Int32. Additional arguments for 'AnyType' types.

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 216] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationEx Overload [▶ 201]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotificationEx [▶ 340]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

6.2.1.2.4 AdsClient.AddDeviceNotificationExAsync Method

Overload List

	Name	Description
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 206]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 208]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217] should always be called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationExAsync Overload [▶ 206]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsClient.AddDeviceNotificationExAsync Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type anyType,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data.
anyType	Type: System.Type Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

AddDeviceNotificationExAsync Overload [▶ 206]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 217]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.AdsNotificationError [▶ 339]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.1.2.5 AdsClient.CleanupSymbolTable Method

Clears the internal symbol cache.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CleanupSymbolTable()
```

Implements

IAdsSymbolicAccess.CleanupSymbolTable. [▶ 942]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.6 AdsClient.Close Method

Closes this AdsClient [▶ 154]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Close()
```

Implements

IConnection.Close. [▶ 78]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.7 AdsClient.Connect Method

Overload List

	Name	Description
	Connect(Int32) [▶ 211]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 211]	Connects the target
	Connect(AmsPort) [▶ 212]	Connects to the local target ADS Device.
	Connect(String, Int32) [▶ 212]	Connects to the target ADS Device.
	Connect(AmsNetId, Int32) [▶ 213]	Connects to the target ADS Device.
	Connect(AmsNetId, AmsPort) [▶ 214]	Connects to the target ADS Device.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (Int32)

Connects to the local target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    int port
)
```

Parameters

port	Type: System.Int32 The port number of the local ADS target device to connect to.

Implements

IAdsConnectAddress.Connect(Int32) [▶ 763]

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (AmsAddress)

Connects the target

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    AmsAddress address
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.

Implements

IAdsConnectAddress.Connect(AmsAddress) [▶ 763]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (AmsPort)

Connects to the local target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    AmsPort port
)
```

Parameters

port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port number of the local ADS target device to connect to.

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (String, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    string netId,
    int port
)
```

Parameters

netId	Type: System.String The AmsNetId [▶ 665] of the ADS target device specified as string.
port	Type: System.Int32 The port number of the ADS target device.

Implements

IAdsConnectAddress.Connect(String, Int32) [▶ 764]

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (AmsNetId, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    AmsNetId netId,
    int port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The AmsNetId of the target device.
port	Type: System.Int32 The Ams Port number on the target device to connect to.

Implements

IAdsConnectAddress.Connect(AmsNetId, Int32) [▶ 764]

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Connect Method (AmsNetId, AmsPort)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Connect(
    AmsNetId netId,
    AmsPort port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The AmsNetId [▶ 665] of the ADS target device specified as string.
port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port number of the ADS target device.

Reference

AdsClient Class [▶ 154]

Connect Overload [▶ 210]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.8 AdsClient.ConnectAndWaitAsync Method

Connects to the target address and waits until the AdsClient [▶ 154] is disconnected asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task ConnectAndWaitAsync(
    AmsAddress address,
    CancellationToken cancel
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The target address.
cancel	Type: System.Threading.CancellationToken Cancellation Token.

Return Value

Type: Task
Returns a task object that represents the ConnectAndWaitAsync(AmsAddress, CancellationToken) operation as result.

Remarks

This method is used for scenarios, where the AdsClient [▶ 154] disconnects from other code asynchronously. When this method returns, the connection is already terminated and only additional cleanup code should be processed.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.9 AdsClient.CreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint CreateVariableHandle(
    string symbolPath
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.

Return Value

Type: UInt32
The symbols/variable handle

Implements

IAdsHandle.CreateVariableHandle(String) [▶ 829]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandle(String) is the DeleteVariableHandle(UInt32) [▶ 218]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.DeleteVariableHandle(UInt32) [▶ 218]

AdsClient.CreateVariableHandleAsync(String, CancellationToken) [▶ 216]

AdsClient.TryCreateVariableHandle(String, UInt32.) [▶ 281]

6.2.1.2.10 AdsClient.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> CreateVariableHandleAsync(
    string symbolPath,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'CreateVariableHandle' operation. The ResultHandle [▶ 1005] parameter contains the variable handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandleAsync(String, CancellationToken) is the DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 219]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 219]

AdsClient.TryCreateVariableHandle(String, UInt32.) [▶ 281]

AdsClient.CreateVariableHandle(String) [▶ 215]

6.2.1.2.11 AdsClient.DeleteDeviceNotification Method

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void DeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.

Implements

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

Remarks

This is the complementary method to AddDeviceNotification Overload [▶ 844] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

AdsClient.AdsNotification [▶ 338]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.1.2.12 AdsClient.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'DeleteDeviceNotification' operation. The ErrorCode [▶ 992] property contains the ADS error code after execution.

Implements

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

This is the complementary method to AddDeviceNotificationAsync Overload [▶ 847] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotificationAsync Overload [▶ 847]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotification Overload [▶ 844]

6.2.1.2.13 AdsClient.DeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void DeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Return Value

Type:
The ADS error code.

Implements

IAdsHandle.DeleteVariableHandle(UInt32) [▶ 830]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) [▶ 285] is the TryCreateVariableHandle(String, UInt32.) [▶ 281]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.CreateVariableHandle(String) [▶ 215]

AdsClient.TryDeleteVariableHandle(UInt32) [▶ 285]

AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 219]

6.2.1.2.14 AdsClient.DeleteVariableHandleAsync Method

Releases the specified symbol/variable handle asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'DeleteVariableHandle' operation. The ResultAds [▶ 989] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this DeleteVariableHandleAsync(UInt32, CancellationToken) is the CreateVariableHandleAsync(String, CancellationToken) [▶ 216]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.CreateVariableHandleAsync(String, CancellationToken) [▶ 216]

AdsClient.TryDeleteVariableHandle(UInt32) [▶ 285]

AdsClient.DeleteVariableHandle(UInt32) [▶ 218]

6.2.1.2.15 AdsClient.Disconnect Method

Disconnects this AdsClient [▶ 154] from the local ADS router.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Disconnect()
```

Return Value

Type: Boolean
true if disconnected, false otherwise.

Implements

IConnection.Disconnect. [▶ 79]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.16 AdsClient.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.17 AdsClient.Finalize Method

Finalizes an instance of the AdsClient [▶ 154] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override void Finalize()
```

Implements

Object.Finalize.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.18 AdsClient.InvokeRpcMethod Method

Overload List

	Name	Description
	InvokeRpcMethod(String, String, .Object.) [▶ 222]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 223]	Invokes the specified RPC Method

	Name	Description
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 225]	Invokes the specified RPC Method

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethod Method (String, String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object.) [▶ 890]

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

InvokeRpcMethod Overload [▶ 221]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethod Method (String, String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL
outParameters	Type: .System.Object.. The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);
```

```
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

InvokeRpcMethod Overload [▶ 221]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifiying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifiying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
```

```
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

InvokeRpcMethod Overload [▶ 221]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.19 AdsClient.InvokeRpcMethodAsync Method

Overload List

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 228]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 230]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 231]	invoke RPC method as an asynchronous operation.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

InvokeRpcMethodAsync Overload [▶ 227]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [▶ 1028] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

InvokeRpcMethodAsync Overload [▶ 227]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

invoke RPC method as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod rpcMethod,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpec,
    AnyTypeSpecifier returnSpec,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The symbol.
rpcMethod	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The RPC method.
inParameters	Type: .System.Object. The in parameters.
outSpec	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out spec.
returnSpec	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The return spec.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
Task<ResultRpcMethod>.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	
ArgumentNullException	symbol
ArgumentNullException	rpcMethod

Reference

AdsClient Class [▶ 154]

InvokeRpcMethodAsync Overload [▶ 227]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.20 AdsClient.Read Method

Overload List

	Name	Description
	Read(UInt32, Memory) [▶ 233]	
	Read(UInt32, UInt32, Memory) [▶ 233]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Read Method (UInt32, Memory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Read(
    uint variableHandle,
    Memory readBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory

Return Value

Type: Int32

Reference

AdsClient Class [▶ 154]

Read Overload [▶ 233]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Read Method (UInt32, UInt32, Memory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory

Return Value

Type: Int32

Reference

AdsClient Class [▶ 154]

Read Overload [▶ 233]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.21 AdsClient.ReadAny Method

Overload List

	Name	Description
	ReadAny.T.(UInt32) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 235]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type) [▶ 236]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 237]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 238]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 239]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 240]	Reads data synchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 241]	Reads data synchronously from an ADS device and writes it to an object.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny.T. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32) [▶ 707]

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny.T. Method (UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, .Int32.) [▶ 708]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny Method (UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint variableHandle,
    Type type
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read object.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type) [▶ 709]

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny.T. Method (UInt32, UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, UInt32) [▶ 710]

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny Method (UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type, .Int32.) [▶ 710]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny.T. Method (UInt32, UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny Method (UInt32, UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type) [▶ 712]

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAny Method (UInt32, UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]

Exceptions

Exception	Condition
ObjectDisposedException	

Exception	Condition
ClientNotConnectedException [▶ 63]	

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAny Overload [▶ 234]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.22 AdsClient.ReadAnyAsync Method

Overload List

	Name	Description
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 243]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 244]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 245]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 246]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 247]	Reads the value of an Anytype (Primitive type) asynchronously.

	Name	Description
	ReadAnyAsync.T. (UInt32, UInt32, .Int32., CancellationToken) [▶ 247]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 249]	Reads the value of an Anytype (Primitive type) asynchronously.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync.T. Method (UInt32, CancellationToken)

Reads data synchronously from an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable/symbol handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The Type of the value to be read.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]

Remarks

As object types only primitive types are supported.

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync.T. Method (UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	Type of the object to be read

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync Method (UInt32, Type, CancellationToken)

Reads the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
type	Type: System.Type The type as AnyType (primitive types).
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue [▶ 998] parameter contains the value Value [▶ 1000] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync.T. Method (UInt32, UInt32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T

Return Value

Type: Task.ResultValue [▶ 1029].T..
The asynchronous result.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)

Reads the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
type	Type: System.Type The type as AnyType (primitive types).
args	Type: .System.Int32. The type arguments (AnyType)
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue [▶ 998] parameter contains the value Value [▶ 1000] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync.T. Method (UInt32, UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The type of the result value.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
type	Type: System.Type The type as AnyType (primitive types).
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue [▶ 998] parameter contains the value Value [▶ 1000] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)

Reads the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
type	Type: System.Type The type as AnyType (primitive types).
args	Type: .System.Int32. The type arguments (AnyType)
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous 'ReadState' operation. The ResultAnyValue [▶ 998] parameter contains the value Value [▶ 1000] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]

Reference

AdsClient Class [▶ 154]

ReadAnyAsync Overload [▶ 242]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.23 AdsClient.ReadAnyString Method

Overload List

	Name	Description
	ReadAnyString(UInt32, Int32, Encoding) [▶ 251]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 251]	Reads as string from a specified address.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyString Method (UInt32, Int32, Encoding)

Reads a string from the specified symbol/variable.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
The string value.

Implements

IAdsAnyAccess.ReadAnyString(UInt32, Int32, Encoding) [▶ 722]

Reference

AdsClient Class [▶ 154]

ReadAnyString Overload [▶ 250]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)

Reads as string from a specified address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
System.String.

Implements

IAdsAnyAccess.ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

ReadAnyString Overload [▶ 250]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.24 AdsClient.ReadAnyStringAsync Method

Overload List

	Name	Description
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 253]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 254]	Reads a string from a specified address asynchronously.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read string (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]

Reference

AdsClient Class [▶ 154]

ReadAnyStringAsync Overload [▶ 253]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

Reads a string from a specified address asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]

Reference

AdsClient Class [▶ 154]

ReadAnyStringAsync Overload [▶ 253]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.25 AdsClient.ReadAsync Method

Overload List

	Name	Description
	ReadAsync(UInt32, Memory, Void) [▶ 255]	
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 256]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAsync Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

AdsClient Class [▶ 154]

ReadAsync Overload [▶ 255]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadAsync Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

AdsClient Class [▶ 154]

ReadAsync Overload [▶ 255]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.26 AdsClient.ReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ReadDataType(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the data type (without namespace)

Return Value

Type: IDataType [▶ 1986]
An containing the requested type.

Implements

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [▶ 943]

6.2.1.2.27 AdsClient.ReadDataTypeAsync Method

read data type as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```

Parameters

typeName	Type: System.String Name of the data type.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IDataType [▶ 1986]..
A task that represents the asynchronous 'ReadDataType' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [▶ 943]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

6.2.1.2.28 AdsClient.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DeviceInfo ReadDeviceInfo()
```

Return Value

Type: DeviceInfo [▶ 698]
DeviceInfo struct containing the name of the device and the version information.

Implements

IAdsConnection.ReadDeviceInfo. [▶ 795]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.29 AdsClient.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultDeviceInfo [▶ 1001].
A task that represents the asynchronous 'ReadDeviceState' operation. The ResultDeviceInfo [▶ 1001] parameter contains the value DeviceInfo [▶ 1003] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsConnection.ReadDeviceInfoAsync(CancellationToken) [▶ 796]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.30 AdsClient.ReadState Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo ReadState()
```

Return Value

Type: StateInfo [▶ 1041]
The ADS statue and device status.

Implements

IAdsStateProvider.ReadState. [▶ 931]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.31 AdsClient.ReadStateAsync Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultReadDeviceState [▶ 1016].
A task that represents the asynchronous 'ReadState' operation. The ResultReadDeviceState [▶ 1016] parameter contains the state (State [▶ 1018]) as long as the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.ReadStateAsync(CancellationToken) [▶ 931]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.32 AdsClient.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IAdsSymbol ReadSymbol(
    string name
)
```

Parameters

name	Type: System.String Name of the symbol.

Return Value

Type: IAdsSymbol [▶ 1379]
A IAdsSymbol2 containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.ReadSymbol(String) [▶ 943]

Exceptions

Exception	Condition
AdsErrorException [▶ 583]	Thrown when the ADS call fails.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.33 AdsClient.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(
    string name,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IAdsSymbol [▶ 1379]..
A task that represents the asynchronous 'ReadSymbolInfo' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken) [▶ 944]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.34 AdsClient.ReadValue Method

Overload List

	Name	Description
	ReadValue.T.(String) [▶ 263]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue(ISymbol) [▶ 263]	Reads the value of a symbol and returns it as an object.
	ReadValue.T.(ISymbol) [▶ 264]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 265]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValue.T. Method (String)

Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadValue<T>(
    string name
)
```

Parameters

name	Type: System.String Name of the ADS symbol.

Type Parameters

T	The value type

Return Value

Type: T
Value of the symbol

Implements

IAdsSymbolicAccess.ReadValue.T.(String) [▶ 945]

Reference

AdsClient Class [▶ 154]

ReadValue Overload [▶ 262]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValue Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Return Value

Type: Object
The value of the symbol as an object.

Implements

IAdsSymbolicAccess.ReadValue(ISymbol) [▶ 946]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsClient Class [▶ 154]

ReadValue Overload [▶ 262]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValue.T. Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Type Parameters

T	The value type.

Return Value

Type: T
The value of the symbol.

Implements

IAdsSymbolicAccess.ReadValue.T.(ISymbol) [▶ 946]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsClient Class [▶ 154]

ReadValue Overload [▶ 262]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValue Method (String, Type)

Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    string name,
    Type type
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.

Return Value

Type: Object
Value of the symbol

Implements

IAdsSymbolicAccess.ReadValue(String, Type) [▶ 947]

Exceptions

Exception	Condition
ObjectDisposedException	

Exception	Condition
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

ReadValue Overload [▶ 262]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.35 AdsClient.ReadValueAsync Method

Overload List

	Name	Description
	ReadValueAsync.TValue.(String, CancellationToken) [▶ 266]	Reads the value of a symbol asynchronously.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 267]	Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync.TValue.(ISymbol, CancellationToken) [▶ 268]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync(String, Type, CancellationToken) [▶ 269]	Reads the value of a symbol asynchronously.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValueAsync.TValue. Method (String, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<TValue>> ReadValueAsync<TValue>(
    string name,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

TValue	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].TValue..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value) and the ErrorCode [▶ 992] after execution.

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [▶ 154]

ReadValueAsync Overload [▶ 266]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValueAsync Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object. Strings and all primitive data types (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
The ResultAnyValue [▶ 998] as customized task object.

Implements

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [▶ 949]

Reference

AdsClient Class [▶ 154]

ReadValueAsync Overload [▶ 266]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValueAsync.TValue. Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<TValue>> ReadValueAsync<TValue>(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

TValue	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].TValue..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value) and the ErrorCode [▶ 992] after execution.

Exceptions

Exception	Condition
NotImplementedException	

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsClient Class [▶ 154]

ReadValueAsync Overload [▶ 266]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadValueAsync(
    string name,
    Type type,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [▶ 951]

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [▶ 154]

ReadValueAsync Overload [▶ 266]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.36 AdsClient.ReadWrite Method

Overload List

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 270]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 271]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadWrite Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

AdsClient Class [▶ 154]

ReadWrite Overload [▶ 270]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadWrite Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

AdsClient Class [▶ 154]

ReadWrite Overload [▶ 270]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.37 AdsClient.ReadWriteAsync Method

Overload List

	Name	Description
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 272]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 273]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadWriteAsync Method (UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

AdsClient Class [▶ 154]

ReadWriteAsync Overload [▶ 272]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.ReadWriteAsync Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

AdsClient Class [▶ 154]

ReadWriteAsync Overload [▶ 272]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.38 AdsClient.RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged [▶ 340] events as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be registered for AdsStateChanged calls.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.RegisterAdsStateChangedAsync(EventHandler.AdsStateChangedEventArgs., CancellationToken) [▶ 932]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.39 AdsClient.RegisterSymbolVersionChangedAsync Method

Registers for an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to register.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.RegisterSymbolVersionChangedAsync(EventHandler.AdsSymbolVersionChangedEventArgs., CancellationToken) [▶ 935]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.40 AdsClient.TryAddDeviceNotification Method

Overload List

	Name	Description
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 276]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 277]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 338] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data.
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS ErrorCode.

Implements

IAdsNotifications.TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 283] should always be called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

TryAddDeviceNotification Overload [▶ 275]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotification [▶ 338]

AdsClient.TryDeleteDeviceNotification(UInt32) [▶ 283]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsClient.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsNotifications.TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 338] as value.
Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 864] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

TryAddDeviceNotification Overload [▶ 275]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [▶ 864]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.1.2.41 AdsClient.TryAddDeviceNotificationEx Method

Overload List

	Name	Description
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 279]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 280]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 340] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')
handle	Type: System.UInt32. The notification handle

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 283] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

TryAddDeviceNotificationEx Overload [▶ 278]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotificationEx [▶ 340]

AdsClient.DeleteDeviceNotification(UInt32) [▶ 216]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsClient.TryAddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type anyType,
    int[] args,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
anyType	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. The 'AnyType' arguments.
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error code.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

AdsClient Class [▶ 154]

TryAddDeviceNotificationEx Overload [▶ 278]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.1.2.42 AdsClient.TryCreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryCreateVariableHandle(
    string symbolPath,
    out uint variableHandle
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.
variableHandle	Type: System.UInt32. The symbols handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [▶ 834]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryCreateVariableHandle(String, UInt32.) is the TryDeleteVariableHandle(UInt32) [▶ 285]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryDeleteVariableHandle(UInt32) [▶ 285]

AdsClient.CreateVariableHandleAsync(String, CancellationToken) [▶ 216]

AdsClient.CreateVariableHandle(String) [▶ 215]

6.2.1.2.43 AdsClient.TryDeleteDeviceNotification Method

Overload List

	Name	Description
	TryDeleteDeviceNotification(UInt32) [▶ 283]	Deletes a registered notification.
	TryDeleteDeviceNotification(UInt32, Int32) [▶ 284]	Deletes a registered notification.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryDeleteDeviceNotification Method (UInt32)

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [▶ 864]

Remarks

This is the complementary method to TryAddDeviceNotification Overload [▶ 859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsClient Class [▶ 154]

TryDeleteDeviceNotification Overload [▶ 282]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

AdsClient.TryDeleteDeviceNotification Method (UInt32, Int32)

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle,
    int timeout
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.
timeout	Type: System.Int32 The timeout.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

This is the complementary method to TryAddDeviceNotification Overload [▶ 859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsClient Class [▶ 154]

TryDeleteDeviceNotification Overload [▶ 282]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.1.2.44 AdsClient.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsHandle.TryDeleteVariableHandle(UInt32) [▶ 835]

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32.) [▶ 281]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryCreateVariableHandle(String, UInt32.) [▶ 281]

AdsClient.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 219]

AdsClient.DeleteVariableHandle(UInt32) [▶ 218]

6.2.1.2.45 AdsClient.TryInvokeRpcMethod Method

Overload List

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 286]	Invokes the specified RPC Method

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 288]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 290]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 291]	Tries the invoke RPC method.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryInvokeRpcMethod Method (String, String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.

inParameters	Type: .System.Object. The parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;
```

```
        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

TryInvokeRpcMethod Overload [▶ 285]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryInvokeRpcMethod Method (String, String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

TryInvokeRpcMethod Overload [▶ 285]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol.
methodName	Type: System.String Name of the method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsClient Class [▶ 154]

TryInvokeRpcMethod Overload [▶ 285]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryInvokeRpcMethod Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Tries the invoke RPC method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    IRpcCallableInstance symbol,
    IRpcMethod rpcMethod,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpec,
    AnyTypeSpecifier returnSpec,
    out Object[] outParameters,
    out Object returnValue
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The symbol.
rpcMethod	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The RPC method.
inParameters	Type: .System.Object. The in parameters.
outSpec	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out spec.
returnSpec	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The return spec.
outParameters	Type: .System.Object.. The out parameters.
returnValue	Type: System.Object. The return value.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	
ArgumentNullException	symbol
ArgumentNullException	rpcMethod

Reference

AdsClient Class [▶ 154]

TryInvokeRpcMethod Overload [▶ 285]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.46 AdsClient.TryRead Method

Overload List

	Name	Description
	TryRead(UInt32, Memory, Void) [▶ 293]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 294]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryRead Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryRead Overload [▶ 293]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryRead Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory memory,
    void readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
memory	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryRead Overload [▶ 293]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.47 AdsClient.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)
```

Parameters

typeName	Type: System.String Name of the symbol.
dataType	Type: TwinCAT.TypeSystem.IDataType [▶ 1986]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
A IDataType [▶ 1986] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

6.2.1.2.48 AdsClient.TryReadState Method

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode [▶ 575] of the ADS read state call. Check for NoError [▶ 575] to see if call was successful.

Implements

IAdsStateProvider.TryReadState(StateInfo.) [▶ 932]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.49 AdsClient.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadSymbol(
    string name,
    out IAdsSymbol symbol
)
```

Parameters

name	Type: System.String Name of the symbol.
symbol	Type: TwinCAT.Ads.TypeSystem.IAdsSymbol [▶ 1379]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
A IAdsSymbol [▶ 1379] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol.) [▶ 953]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.50 AdsClient.TryReadValue Method

Overload List

	Name	Description
	TryReadValue.T.(String, T.) [▶ 297]	Reads the value of a symbol and returns the value as object.
	TryReadValue(ISymbol, Object.) [▶ 298]	Reads the value of a symbol and returns it as an object.
	TryReadValue.T.(ISymbol, T.) [▶ 299]	Reads the value of a symbol and returns it as an object.
	TryReadValue(String, Type, Object.) [▶ 300]	Reads the value of a symbol and returns the value as object.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadValue.T. Method (String, T.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue<T>(
    string name,
    out T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T. The read value of the Symbol.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Implements

IAdsSymbolicAccess.TryReadValue.T.(String, T.) [▶ 954]

Exceptions

Exception	Condition
NotImplementedException	

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [▶ 154]

TryReadValue Overload [▶ 297]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadValue Method (ISymbol, Object.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue(
    ISymbol symbol,
    out Object value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: System.Object. The value.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [▶ 954]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsClient Class [▶ 154]

TryReadValue Overload [▶ 297]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadValue.T. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: T. The value.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.) [▶ 955]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsClient Class [▶ 154]

TryReadValue Overload [▶ 297]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue(
    string name,
    Type type,
    out Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.
value	Type: System.Object. The read value of the Symbol.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Implements

IAdsSymbolicAccess.TryReadValue(String, Type, Object.) [▶ 956]

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsClient Class [▶ 154]

TryReadValue Overload [▶ 297]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.51 AdsClient.TryReadWrite Method

Overload List

	Name	Description
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 301]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 302]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadWrite Method (UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryReadWrite Overload [▶ 301]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryReadWrite Overload [▶ 301]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.52 AdsClient.TryResurrect Method

Resurrects the connection

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryResurrect(
    out AdsException error
)
```

Parameters

error	Type: TwinCAT.AdsException [▶ 57]. The error.

Return Value

Type: Boolean
true if resurrection was accepted, false otherwise.

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.53 AdsClient.TryWrite Method

Overload List

	Name	Description
	TryWrite(UInt32, ReadOnlyMemory) [▶ 303]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 304]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWrite Method (UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryWrite Overload [▶ 303]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWrite Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryWrite Overload [▶ 303]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.54 AdsClient.TryWriteControl Method

Overload List

	Name	Description
	TryWriteControl(StateInfo) [▶ 305]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 306]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsStateControl.TryWriteControl(StateInfo) [▶ 917]

Reference

AdsClient Class [▶ 154]

TryWriteControl Overload [▶ 305]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsClient Class [▶ 154]

TryWriteControl Overload [▶ 305]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.55 AdsClient.TryWriteValue Method

Overload List

	Name	Description
	TryWriteValue(String, Object) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(String, T) [▶ 307]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 308]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

	Name	Description
	TryWriteValue.T. (ISymbol, T) [▶ 309]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue(String, Object) [▶ 957]

Reference

AdsClient Class [▶ 154]

TryWriteValue Overload [▶ 306]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue<T>(
    string name,
    T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue.T.(String, T) [▶ 958]

Reference

AdsClient Class [▶ 154]

TryWriteValue Overload [▶ 306]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [▶ 959]

Reference

AdsClient Class [▶ 154]

TryWriteValue Overload [▶ 306]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.TryWriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [▶ 960]

Reference

AdsClient Class [▶ 154]

TryWriteValue Overload [▶ 306]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.56 AdsClient.UnregisterAdsStateChangedAsync Method

unregister ads state changed as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> UnregisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be unregistered.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.UnregisterAdsStateChangedAsync(EventHandler.AdsStateChangedEventArgs., CancellationToken) [▶ 933]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.57 AdsClient.UnregisterSymbolVersionChangedAsync Method

Unregisters from an AdsSymbolVersionChanged [▶ 342] event as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to unregister.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.UnregisterSymbolVersionChangedAsync(EventHandler.AdsSymbolVersionChangedEventArgs., CancellationToken) [▶ 936]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.58 AdsClient.Write Method

Overload List

	Name	Description
	Write(UInt32, ReadOnlyMemory) [▶ 312]	

	Name	Description
	Write(UInt32, UInt32) [▶ 312]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 313]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Write Method (UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

AdsClient Class [▶ 154]

Write Overload [▶ 311]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Write Method (UInt32, UInt32)

Triggers a 'Write' call to the ADS device at the specified address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.

Implements

IAdsReadWrite2.Write(UInt32, UInt32) [▶ 878]

Reference

AdsClient Class [▶ 154]

Write Overload [▶ 311]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

AdsClient Class [▶ 154]

Write Overload [▶ 311]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.59 AdsClient.WriteAny Method

Overload List

	Name	Description
	WriteAny(UInt32, Object) [▶ 314]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 315]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 315]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 316]	Writes an object synchronously to an ADS device.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAny Method (UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint variableHandle,
    Object value
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.

Implements

IAdsAnyAccess.WriteAny(UInt32, Object) [▶ 726]

Reference

AdsClient Class [▶ 154]

WriteAny Overload [▶ 314]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAny Method (UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Implements

IAdsAnyAccess.WriteAny(UInt32, Object, .Int32.) [▶ 727]

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

WriteAny Overload [▶ 314]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAny Method (UInt32, UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.

Implements

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object) [▶ 728]

Reference

AdsClient Class [▶ 154]

WriteAny Overload [▶ 314]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAny Method (UInt32, UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.

value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Implements

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]

Remarks

If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

WriteAny Overload [▶ 314]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.60 AdsClient.WriteAnyAsync Method

Overload List

	Name	Description
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 318]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 319]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 320]	Write the value of an Anytype (Primitive type) asynchronously.

	Name	Description
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 321]	Write the value of an Anytype (Primitive type) asynchronously.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyAsync Method (UInt32, Object, CancellationToken)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].

Type of value Parameter	Necessary Arguments (args)
string[]	
Array	

Reference

AdsClient Class [▶ 154]

WriteAnyAsync Overload [▶ 317]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyAsync Method (UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsClient Class [▶ 154]

WriteAnyAsync Overload [▶ 317]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)

Write the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'ReadState' operation. The ResultWrite [▶ 1032] parameter contains the value the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]

Reference

AdsClient Class [▶ 154]

WriteAnyAsync Overload [▶ 317]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)

Write the value of an Anytype (Primitive type) asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
value	Type: System.Object The value.
args	Type: .System.Int32. The type arguments (AnyType)
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'ReadState' operation. The ResultWrite [▶ 1032] parameter contains the value the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]

Reference

AdsClient Class [▶ 154]

WriteAnyAsync Overload [▶ 317]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.61 AdsClient.WriteAnyStringAsync Method

Overload List

	Name	Description
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 322]	Writes the string (Potentially unsafe!)
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 323]	Writes the string (Potentially unsafe!)

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyStringAsync Method (String, String, Int32, Encoding, CancellationToken)

Writes the string (Potentially unsafe!)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyStringAsync(
    string symbolPath,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol path.
value	Type: System.String The value.
length	Type: System.Int32 The length of the string to write
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsClient Class [▶ 154]

WriteAnyStringAsync Overload [▶ 322]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAnyStringAsync Method (UInt32, String, Int32, Encoding, CancellationToken)

Writes the string (Potentially unsafe!)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyStringAsync(
    uint variableHandle,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
value	Type: System.String The value.
length	Type: System.Int32 The length of the string to write
encoding	Type: System.Text.Encoding The encoding.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsClient Class [▶ 154]

WriteAnyStringAsync Overload [▶ 322]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.62 AdsClient.WriteAsync Method

Overload List

	Name	Description
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 324]	
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 325]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 326]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAsync Method (UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

AdsClient Class [▶ 154]

WriteAsync Overload [▶ 324]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAsync Method (UInt32, UInt32, CancellationToken)

Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'ReadWrite' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

AdsClient Class [▶ 154]

WriteAsync Overload [▶ 324]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteAsync Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

AdsClient Class [▶ 154]

WriteAsync Overload [▶ 324]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.63 AdsClient.WriteControl Method

Overload List

	Name	Description
	WriteControl(StateInfo) [▶ 327]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 328]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Implements

IAdsStateControl.WriteControl(StateInfo) [▶ 918]

Reference

AdsClient Class [▶ 154]

WriteControl Overload [▶ 327]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Reference

AdsClient Class [▶ 154]

WriteControl Overload [▶ 327]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.64 AdsClient.WriteControlAsync Method

Overload List

	Name	Description
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 328]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 329]	

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> WriteControlAsync(
    AdsState adsState,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

adsState	Type: TwinCAT.Ads.AdsState [▶ 626] The ADS state.
deviceState	Type: System.UInt16 The device state.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'WriteControl' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateControl.WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]

Reference

AdsClient Class [▶ 154]

WriteControlAsync Overload [▶ 328]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> WriteControlAsync(
    AdsState adsState,
    ushort deviceState,
    ReadOnlyMemory writeData,
    void cancel
)
```

Parameters

adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
writeData	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultAds [▶ 989].

Reference

AdsClient Class [▶ 154]

WriteControlAsync Overload [▶ 328]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.65 AdsClient.WriteSymbolAsync Method

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteSymbolAsync(
    string name,
    Object value,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.66 AdsClient.WriteValue Method

Overload List

	Name	Description
	WriteValue(String, Object) [▶ 331]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue.T.(String, T) [▶ 332]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 332]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(ISymbol, T) [▶ 333]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Implements

IAdsSymbolicAccess.WriteValue(String, Object) [▶ 961]

Reference

AdsClient Class [▶ 154]

WriteValue Overload [▶ 331]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue<T>(
    string name,
    T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	the value type.

Implements

IAdsSymbolicAccess.WriteValue.T.(String, T) [▶ 961]

Reference

AdsClient Class [▶ 154]

WriteValue Overload [▶ 331]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Implements

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [▶ 962]

Reference

AdsClient Class [▶ 154]

WriteValue Overload [▶ 331]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Implements

IAdsSymbolicAccess.WriteValue.T.(ISymbol, T) [▶ 963]

Reference

AdsClient Class [▶ 154]

WriteValue Overload [▶ 331]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.2.67 AdsClient.WriteValueAsync Method

Overload List

	Name	Description
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 334]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 335]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 336]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValueAsync.T. Method (String, T, CancellationToken)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync<T>(
    string name,
    T value,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]

Reference

AdsClient Class [▶ 154]

WriteValueAsync Overload [▶ 334]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValueAsync Method (ISymbol, Object, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]

Reference

AdsClient Class [▶ 154]

WriteValueAsync Overload [▶ 334]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.WriteValueAsync.T. Method (ISymbol, T, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]

Reference

AdsClient Class [▶ 154]

WriteValueAsync Overload [▶ 334]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3 AdsClient Events

The AdsClient [▶ 154] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 338]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 339]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 340]	Occurs when the ADS devices sends a notification to the client.
	AdsStateChanged [▶ 340]	Occurs when the ADS state changes.
	AdsSumNotification [▶ 341]	Occurs when Notifications are send (bundled notifications)
	AdsSymbolVersionChanged [▶ 342]	Occurs when the symbol version has been changed changes.
	ConnectionStateChanged [▶ 343]	Occurs when the connection state has been changed.
	RouterStateChanged [▶ 344]	Occurs when the state of the local Router has changed.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.1 AdsClient.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationEventArgs> AdsNotification
```

Value

Type: System.EventHandler.AdsNotificationEventArgs [▶ 597].

Implements

IAdsNotifications.AdsNotification [▶ 866]

Remarks

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

Examples

Example of receiving AdsNotification events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification2;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];
    int size = sizeof(UInt32);

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new Notific
ationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
```

```
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.2 AdsClient.AdsNotificationError Event

Occurs when a exception has occurred during notification management.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError
```

Value

Type: System.EventHandler.AdsNotificationErrorEventArgs [▶ 595].

Implements

IAdsNotifications.AdsNotificationError [▶ 867]

Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the AdsInvalidNotificationException [▶ 591] which describes the invalid notification handle by its Handle [▶ 593] property.

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.AdsInvalidNotificationException [▶ 591]

AdsClient.AdsNotification [▶ 338]

AdsClient.AdsNotificationEx [▶ 340]

6.2.1.3.3 AdsClient.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

Value

Type: System.EventHandler.AdsNotificationExEventArgs [▶ 601].

Implements

IAdsNotifications.AdsNotificationEx [▶ 868]

Remarks

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshallers.

Examples

Example of receiving AdsNotificationEx events.

Trigger on changed values by ADS Notifications

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new Notifica
tionSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint),null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle,ca
ncel); // Unregister Event
}
```

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.4 AdsClient.AdsStateChanged Event

Occurs when the ADS state changes.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

Value

Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627].

Implements

IAdsStateProvider.AdsStateChanged [▶ 934]

Remarks

This works only for ports that support Notifications (e.g. Port 851 but not Port 10000). In case of

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.5 AdsClient.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```

Value

Type: System.EventHandler.AdsSumNotificationEventArgs.

Implements

IAdsNotifications.AdsSumNotification [▶ 868]

Remarks

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The AdsNotification [▶ 338] and

AdsNotificationEx [▶ 340] events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification event. The full performance is reached only, when all notifications are handled on this event.

Examples

Example of receiving AdsSumNotification events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsSumNotification += Client_SumNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), n
ew NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;

    foreach(Notification notification in notifications)
    {
    // Notifications can be handled more efficiently, because they occur togeterh
    // handler and can be transformed/
synchronized in one step compared to AdsClient.AdsNotifcation events.
    }
}
```

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

AdsClient.AdsNotification [▶ 338]

6.2.1.3.6 AdsClient.AdsSymbolVersionChanged Event

Occurs when the symbol version has been changed changes.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

Value

Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637].

Implements

IAdsSymbolChangedProvider.AdsSymbolVersionChanged [▶ 937]

Remarks

This is the case when the connected ADS server restarts. This invalidates all actual opened symbol handles. The SymbolVersion counter doesn't trigger, when an online change is made on the PLC (ports 801, ..., 851 ...)

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.7 AdsClient.ConnectionStateChanged Event

Occurs when the connection state has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: System.EventHandler.ConnectionStateChangedEventArgs [▶ 68].

Implements

IConnectionStateProvider.ConnectionStateChanged [▶ 86]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.1.3.8 AdsClient.RouterStateChanged Event

Occurs when the state of the local Router has changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```

Value

Type: System.EventHandler.AmsRouterNotificationEventArgs [▶ 695].

Implements

IRouterNotificationProvider.RouterStateChanged [▶ 973]

Reference

AdsClient Class [▶ 154]

TwinCAT.Ads Namespace [▶ 151]

6.2.2 AdsClientSettings Class

Settings object for the AdsClient [▶ 154] class.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsClientSettings
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsClientSettings
```

The AdsClientSettings type exposes the following members.

Constructors

	Name	Description
	AdsClientSettings [▶ 345]	Creates a Default settings AdsClientSettings object with custom timeout.

Properties

	Name	Description
	CompatibilityDefault [▶ 346]	Compatibility settings object
	Default [▶ 347]	Gets the default settings (Default interceptors, Timout 5000 ms)
	FastWriteThrough [▶ 347]	Gets a Settings object that configures the AdsClient for FastWriteThrough
	Timeout [▶ 348]	The communication Timeout that is set initially on the AdsClient [▶ 154]

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This AdsClientSettings object is used to initalize the AdsClient [▶ 154] with application appropriate settings. Several predefined application dependant settings are available as static properties:

- Default [▶ 347]
- FastWriteThrough [▶ 347]
- CompatibilityDefault [▶ 346]

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.2.1 AdsClientSettings Constructor

Creates a Default settings AdsClientSettings [▶ 344] object with custom timeout.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsClientSettings(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout of the AdsClient [▶ 154] in milliseconds.

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.2 AdsClientSettings Properties

The AdsClientSettings [▶ 344] type exposes the following members.

Properties

	Name	Description
	CompatibilityDefault [▶ 346]	Compatibility settings object
	Default [▶ 347]	Gets the default settings (Default interceptors, Timout 5000 ms)
	FastWriteThrough [▶ 347]	Gets a Settings object that configures the AdsClient for FastWriteThrough
	Timeout [▶ 348]	The communication Timeout that is set initially on the AdsClient [▶ 154]

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.2.1 AdsClientSettings.CompatibilityDefault Property

Compatibility settings object

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsClientSettings CompatibilityDefault { get; }
```

Property Value

Type: AdsClientSettings [▶ 344]
The settings object.

Remarks

The compatibility settings intitialize the AdsClient the same way as it is done in earlier versions of the TwinCAT.Ads.dll (earlier than Version 4.2)

- All [▶ 1053]
- No FailFastHandlerInterceptor active.
- Default communicationtimeout 5000ms.
- Synchronized Notifications.

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.2.2 AdsClientSettings.Default Property

Gets the default settings (Default interceptors, Timout 5000 ms)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsClientSettings Default { get; }
```

Property Value

Type: AdsClientSettings [▶ 344]
The default.

Remarks

Creates an settings object, with specification for All [▶ 1053] and FailFastHandlerInterceptor.

- All [▶ 1053]
- FailFastHandlerInterceptor is active.
- Default communication timeout 5000ms.
- Not synchronized Notifications.

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.2.3 AdsClientSettings.FastWriteThrough Property

Gets a Settings object that configures the AdsClient for FastWriteThrough

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsClientSettings FastWriteThrough { get; }
```

Property Value

Type: AdsClientSettings [▶ 344]
Client settings for a fast write through (with 200 ms Timeout).

Remarks

The settings typically can be used for polling clients, where the "FailFast" feature will be bypassed. That means, that communication fails doesn't trigger the FailFast interceptor and every Request will go out via ADS. This has the Drawback that communication Timeouts are longer and subsequent timeouts block the ADS mailbox (with the danger of overflows). So use this setting with care for specific purposes and should not be used for standard communication.

- No FailFastHandlerInterceptor active.
- Default communicationtimeout 200ms.
- Not synchronized Notifications.

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.2.4 AdsClientSettings.Timeout Property

The communication Timeout that is set initially on the AdsClient [▶ 154]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Timeout { get; }
```

Property Value

Type: Int32
The timeout.

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.2.3 AdsClientSettings Methods

The AdsClientSettings [▶ 344] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsClientSettings Class [▶ 344]

TwinCAT.Ads Namespace [▶ 151]

6.2.3 AdsCommandId Enumeration

AdsCommandId Enumeration

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AdsCommandId
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Invalid	0	Invalid
	ReadDeviceInfo	1	ReadDeviceInfo command
	Read	2	Read Command
	Write	3	Write Command
	ReadState	4	ReadState Command
	WriteControl	5	WriteControl Command
	AddNotification	6	AddNotification Command
	DeleteNotification	7	DeleteNotification Command
	Notification	8	Notification event.
	ReadWrite	9	ReadWrite Command

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.4 AdsCommunicationStatistics Class

ADS Communication statistics

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsCommunicationStatistics
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsCommunicationStatistics
```

The AdsCommunicationStatistics type exposes the following members.

Properties

	Name	Description
	AccessWaitTime [▶ 352]	Gets the wait time for the next access (Resurrection time) if in Lost [▶ 67].
	ConnectionActiveSince [▶ 352]	Gets the UTC time of the last conenction activation.
	ConnectionEstablishedAt [▶ 353]	Gets the UTC time when the current connection was established.
	ConnectionLostCount [▶ 353]	Gets the connection lost count.
	ConnectionLostTime [▶ 354]	Gets the UTC connection lost time.
	ConnectionResurrections [▶ 354]	Gets the number of resurrections on the AdsConnection [▶ 357]
	ErrorsSinceLastSucceeded [▶ 354]	Gets the error count since last access (UTC)
	LastSucceededAccess [▶ 355]	Gets the UTC time of the last succeeded access.
	Resurrections [▶ 355]	Gets the number of Resurrections of this Session.
	SessionEstablishedAt [▶ 356]	Gets the UTC time when the session was established.

	Name	Description
	TotalCycles [▶ 356]	Gets the total cycles.
	TotalErrors [▶ 357]	Gets the total error count.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

These statistics can be used for communication diagnosis. They contain Error/Succeed counts as well as Resurrection infos.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1 AdsCommunicationStatistics Properties

The AdsCommunicationStatistics [▶ 350] type exposes the following members.

Properties

	Name	Description
	AccessWaitTime [▶ 352]	Gets the wait time for the next access (Resurrection time) if in Lost [▶ 67].
	ConnectionActiveSince [▶ 352]	Gets the UTC time of the last conenction activation.
	ConnectionEstablishedAt [▶ 353]	Gets the UTC time when the current connection was established.
	ConnectionLostCount [▶ 353]	Gets the connection lost count.
	ConnectionLostTime [▶ 354]	Gets the UTC connection lost time.
	ConnectionResurrections [▶ 354]	Gets the number of resurrections on the AdsConnection [▶ 357]

	Name	Description
	ErrorsSinceLastSucceeded [▶ 354]	Gets the error count since last access (UTC)
	LastSucceededAccess [▶ 355]	Gets the UTC time of the last succeeded access.
	Resurrections [▶ 355]	Gets the number of Resurrections of this Session.
	SessionEstablishedAt [▶ 356]	Gets the UTC time when the session was established.
	TotalCycles [▶ 356]	Gets the total cycles.
	TotalErrors [▶ 357]	Gets the total error count.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.1 AdsCommunicationStatistics.AccessWaitTime Property

Gets the wait time for the next access (Resurrection time) if in Lost [▶ 67].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TimeSpan AccessWaitTime { get; }
```

Property Value

Type: TimeSpan
The wait time if in Lost [▶ 67] otherwise **TimeSpan.Zero**.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.2 AdsCommunicationStatistics.ConnectionActiveSince Property

Gets the UTC time of the last conenction activation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ConnectionActiveSince { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
Connection active time.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.3 AdsCommunicationStatistics.ConnectionEstablishedAt Property

Gets the UTC time when the current connection was established.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ConnectionEstablishedAt { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The connection established at.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.4 AdsCommunicationStatistics.ConnectionLostCount Property

Gets the connection lost count.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ConnectionLostCount { get; }
```

Property Value

Type: Int32
The connection lost count.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.5 AdsCommunicationStatistics.ConnectionLostTime Property

Gets the UTC connection lost time.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ConnectionLostTime { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The connection lost time.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.6 AdsCommunicationStatistics.ConnectionResurrections Property

Gets the number of resurrections on the AdsConnection [▶ 357]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ConnectionResurrections { get; }
```

Property Value

Type: Int32
The resurrections.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.7 AdsCommunicationStatistics.ErrorsSinceLastSucceeded Property

Gets the error count since last access (UTC)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ErrorsSinceLastSucceeded { get; }
```

Property Value

Type: Int32
The error count since last access.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.8 AdsCommunicationStatistics.LastSucceededAccess Property

Gets the UTC time of the last succeeded access.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> LastSucceededAccess { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The last succeeded access.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.9 AdsCommunicationStatistics.Resurrections Property

Gets the number of Resurrections of this Session.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Resurrections { get; }
```

Property Value

Type: Int32
The resurrections.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.10 AdsCommunicationStatistics.SessionEstablishedAt Property

Gets the UTC time when the session was established.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset SessionEstablishedAt { get; }
```

Property Value

Type: DateTimeOffset
The session established at.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.11 AdsCommunicationStatistics.TotalCycles Property

Gets the total cycles.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TotalCycles { get; }
```

Property Value

Type: Int32
The total cycles.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.1.12 AdsCommunicationStatistics.TotalErrors Property

Gets the total error count.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TotalErrors { get; }
```

Property Value

Type: Int32
The total error count.

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.4.2 AdsCommunicationStatistics Methods

The AdsCommunicationStatistics [▶ 350] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsCommunicationStatistics Class [▶ 350]

TwinCAT.Ads Namespace [▶ 151]

6.2.5 AdsConnection Class

ADS Connection class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsConnection
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsConnection : IAdsConnection,
    IConnection, IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess, IAdsAnyAccess,
    IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider, IAdsStateControl,
    IAdsSymbolChangedProvider, IAdsRpcInvoke, IAdsReadWriteTimeoutAccess, IAdsStateControlTimeout, I
Disposable
```

The AdsConnection type exposes the following members.

Properties

	Name	Description
	AccessWaitTime [▶ 376]	Gets the access wait time.
	ActiveSince [▶ 376]	Gets the UTC time when tha last active/resurrected Connection was established
	Address [▶ 377]	Gets the AmsAddress [▶ 648] of the ADS server.
	ClientAddress [▶ 377]	Get the AmsAddress [▶ 648] of the ADS client.
	ConnectionEstablishedAt [▶ 378]	Gets the UTC time when the Connection was originally established.
	ConnectionLostCount [▶ 378]	Gets the connection lost count.
	ConnectionLostTime [▶ 378]	Gets the connection lost time.
	ConnectionState [▶ 379]	Gets the current Connection state of the AdsConnection
	DefaultValueEncoding [▶ 380]	Gets the default value encoding.
	Disposed [▶ 381]	Gets a value indicating whether this AdsConnection is disposed.
	Id [▶ 381]	Gets the AdsConnection identifier.
	IsActive [▶ 382]	Gets a value indicating whether communication is in active state
	IsConnected [▶ 382]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	IsLocal [▶ 383]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

	Name	Description
	IsLost [▶ 383]	Gets a value indicating whether the communication is in lost / open state
	IsReconnecting [▶ 384]	Gets a value indicating whether communication is ready for reconnecting
	Name [▶ 384]	Gets the name of this AdsConnection.
	ResurrectingTries [▶ 384]	Gets the number of tries to resurrect the AdsConnection.
	Resurrections [▶ 385]	Gets the number of succeeded connection resurrections.
	Session [▶ 385]	Gets the Session object of the AdsConnection object.
	State [▶ 386]	Gets the current ConnectionState [▶ 379]
	Timeout [▶ 386]	Gets the timeout (in milliseconds)

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 402]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 403]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 405]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 406]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 564] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 408]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 409]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 411]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 412]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 413]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 415]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	CleanupSymbolTable [▶ 416]	Clears the internal symbol cache.
	Close [▶ 417]	Closes the AdsConnection
	Connect [▶ 417]	(Re)Connects the IConnection [▶ 74] when disconnected.
	CreateSymbolLoader [▶ 417]	Creates a new instance of the Symbol loader [▶ 1383] with the specified mode.
	CreateVariableHandle [▶ 429]	Generates a unique handle for an ADS variable.
	CreateVariableHandleAsync [▶ 429]	Determines the Symbol handle by its instance path asynchronously.
	DeleteDeviceNotification [▶ 430]	Deletes an existing notification.

	Name	Description
	DeleteDeviceNotificationAsync [▶ 431]	Deletes a registered notification asynchronously.
	DeleteVariableHandle [▶ 432]	Releases the handle of a ADS variable again.
	DeleteVariableHandleAsync [▶ 432]	Releases the handle of a ADS variable again (asynchronously)
	Disconnect [▶ 434]	Disconnects this IConnection [▶ 74].
	Dispose [▶ 434]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	InvokeRpcMethod(String, String, .Object.) [▶ 435]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 437]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 438]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 441]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 443]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 445]	Invokes the specified RPC Method asynchronously
	Read(UInt32, Memory) [▶ 447]	
	Read(UInt32, UInt32, Memory) [▶ 448]	
	Read(UInt32, UInt32, Memory, Void) [▶ 448]	
	ReadAny(UInt32, Type) [▶ 451]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 453]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 455]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 456]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32., Int32) [▶ 457]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 452]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 454]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 461]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 463]	Reads data synchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 465]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 466]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 459]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 460]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 462]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 464]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyString(UInt32, Int32, Encoding) [▶ 467]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 468]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 470]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 471]	read any string as an asynchronous operation.
	ReadAsync(UInt32, Memory, Void) [▶ 472]	

	Name	Description
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 473]	
	ReadDataType [▶ 473]	Call this method to obtain information about the specified data type.
	ReadDataTypeAsync [▶ 474]	read data type as an asynchronous operation.
	ReadDeviceInfo [▶ 475]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 476]	Reads the identification and version number of an ADS server.
	ReadState. [▶ 477]	Reads the ADS status and the device status from an ADS server.
	ReadState(Int32) [▶ 477]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 478]	Read the ADS State asynchronously
	ReadSymbol [▶ 479]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadSymbolAsync [▶ 479]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 481]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValue(String, Type) [▶ 483]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(String) [▶ 481]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 482]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 485]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync(String, Type, CancellationToken) [▶ 487]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(String, CancellationToken) [▶ 484]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 486]	Reads the value of a symbol asynchronously and returns it as an object.

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 488]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 489]	
	ReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 489]	
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 491]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 491]	
	RegisterAdsStateChangedAsync [▶ 492]	Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.
	RegisterSymbolVersionChangedAsync [▶ 493]	Registers the symbol version changed asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 494]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 495]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 497]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

	Name	Description
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 498]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	TryCreateVariableHandle [▶ 499]	Read (determine) the Symbol handle by its name/path
	TryDeleteDeviceNotification [▶ 500]	Deletes a registered notification.
	TryDeleteVariableHandle [▶ 501]	Releases the specified symbol/variable handle synchronously.
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 503]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 505]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 506]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier.., AnyTypeSpecifier, .Object.., Object.) [▶ 508]	Invokes the rpc method.
	TryRead(UInt32, Memory, Void) [▶ 511]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 511]	

	Name	Description
	TryRead(UInt32, UInt32, Memory, Void, Byte) [▶ 512]	
	TryReadDataType [▶ 513]	Call this method to obtain information about the specified data type.
	TryReadState(StateInfo.) [▶ 514]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadState(Int32, StateInfo.) [▶ 515]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadSymbol [▶ 515]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	TryReadValue(ISymbol, Object.) [▶ 517]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	TryReadValue(String, Type, Object.) [▶ 519]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 517]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 518]	Reads the value of a symbol and returns it as an object.
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 521]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 522]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte, ReadOnlyMemory) [▶ 522]	
	TryWrite(UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 525]	

	Name	Description
	TryWriteControl(StateInfo) [▶ 526]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, Int32) [▶ 527]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 528]	
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 528]	
	TryWriteValue(String, Object) [▶ 529]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 531]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 530]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 532]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	UnregisterAdsStateChangedAsync [▶ 533]	Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.
	UnregisterSymbolVersionChangedAsync [▶ 533]	Unregisters the symbol version changed asynchronous.
	Write(UInt32, ReadOnlyMemory) [▶ 535]	
	Write(UInt32, UInt32) [▶ 535]	Trigger Client Method/Command.
	Write(UInt32, UInt32, Int32) [▶ 536]	Trigger Client Method/Command.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 537]	

	Name	Description
	Write(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 537]	
	WriteAny(UInt32, Object) [▶ 538]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 539]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 540]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 540]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object, .Int32., Int32) [▶ 541]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 542]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 543]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 545]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 545]	write any as an asynchronous operation.
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 547]	write any string as an asynchronous operation.

	Name	Description
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 548]	write any string as an asynchronous operation.
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 549]	
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 550]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 551]	
	WriteControl(StateInfo) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, Int32) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 553]	
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 554]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 555]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 555]	
	WriteSymbolAsync [▶ 556]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(String, Object) [▶ 557]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 559]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

	Name	Description
	WriteValue.T.(String, T) [▶ 558]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue.T.(ISymbol, T) [▶ 560]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 562]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 561]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 563]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Events

	Name	Description
	AdsNotification [▶ 564]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 565]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 566]	Occurs when the ADS devices sends a notification to the client.
	AdsStateChanged [▶ 567]	Occurs when ADS State has been changed.
	AdsSumNotification [▶ 568]	Occurs when Notifications are send (bundled notifications)
	AdsSymbolVersionChanged [▶ 569]	Occurs when the symbol version has been changed.
	ConnectionStateChanged [▶ 570]	Occurs when connection status of the AdsConnection has been changed.
	RouterStateChanged [▶ 571]	Occurs when [router state changed].

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Remarks

The ADS Connection class represents an ADS Point-to-Point Connection between client and server. It is established by using the Connect method of the AdsSession [▶ 603] object. An ADS Connection can have different ConnectionStates [▶ 379], which represent the state of the logical ADS connection.

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.AdsSession [▶ 603]

TwinCAT.Ads.IAdsConnection [▶ 765]

System.IDisposable

6.2.5.1 AdsConnection Properties

The AdsConnection [▶ 357] type exposes the following members.

Properties

	Name	Description
	AccessWaitTime [▶ 376]	Gets the access wait time.
	ActiveSince [▶ 376]	Gets the UTC time when tha last active/resurrected Connection was established
	Address [▶ 377]	Gets the AmsAddress [▶ 648] of the ADS server.
	ClientAddress [▶ 377]	Get the AmsAddress [▶ 648] of the ADS client.
	ConnectionEstablishedAt [▶ 378]	Gets the UTC time when the Connection was originally established.
	ConnectionLostCount [▶ 378]	Gets the connection lost count.
	ConnectionLostTime [▶ 378]	Gets the connection lost time.
	ConnectionState [▶ 379]	Gets the current Connection state of the AdsConnection [▶ 357]
	DefaultValueEncoding [▶ 380]	Gets the default value encoding.
	Disposed [▶ 381]	Gets a value indicating whether this AdsConnection [▶ 357] is disposed.
	Id [▶ 381]	Gets the AdsConnection [▶ 357] identifier.
	IsActive [▶ 382]	Gets a value indicating whether communication is in active state
	IsConnected [▶ 382]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.
	IsLocal [▶ 383]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.
	IsLost [▶ 383]	Gets a value indicating whether the communication is in lost / open state
	IsReconnecting [▶ 384]	Gets a value indicating whether communication is ready for reconnecting
	Name [▶ 384]	Gets the name of this AdsConnection [▶ 357].
	ResurrectingTries [▶ 384]	Gets the number of tries to resurrect the AdsConnection [▶ 357].
	Resurrections [▶ 385]	Gets the number of succeeded connection resurrections.
	Session [▶ 385]	Gets the Session object of the AdsConnection [▶ 357] object.
	State [▶ 386]	Gets the current ConnectionState [▶ 379]
	Timeout [▶ 386]	Gets the timeout (in milliseconds)

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.1 AdsConnection.AccessWaitTime Property

Gets the access wait time.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TimeSpan AccessWaitTime { get; }
```

Property Value

Type: TimeSpan
The access wait time.

Remarks

Gets the Wait Time until the next communication try will be done. This time is calculated as follows: ResurrectionTime - (DateTime.Now - ConnectionLostTime)

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ConnectionLostTime [▶ 378]

SessionSettings.ResurrectionTime [▶ 1039]

6.2.5.1.2 AdsConnection.ActiveSince Property

Gets the UTC time when tha last active/resurrected Connection was established

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ActiveSince { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The active since.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.3 AdsConnection.Address Property

Gets the AmsAddress [▶ 648] of the ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The server address.

Implements

IAdsConnection.Address [▶ 781]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.4 AdsConnection.ClientAddress Property

Get the AmsAddress [▶ 648] of the ADS client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress ClientAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The client address.

Implements

IAdsConnection.ClientAddress [▶ 781]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.5 AdsConnection.ConnectionEstablishedAt Property

Gets the UTC time when the Connection was originally established.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ConnectionEstablishedAt { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The connection established at.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.6 AdsConnection.ConnectionLostCount Property

Gets the connection lost count.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ConnectionLostCount { get; }
```

Property Value

Type: Int32
The connection lost count.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.7 AdsConnection.ConnectionLostTime Property

Gets the connection lost time.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Nullable<DateTimeOffset> ConnectionLostTime { get; }
```

Property Value

Type: Nullable.DateTimeOffset.
The connection lost time.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.8 AdsConnection.ConnectionState Property

Gets the current Connection state of the AdsConnection [▶ 357]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionState ConnectionState { get; }
```

Property Value

Type: ConnectionState [▶ 67]
The state of the connection.

Implements

IConnectionStateProvider.ConnectionState [▶ 85]

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
```

```
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ConnectionStateChanged [▶ 570]

6.2.5.1.9 AdsConnection.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.

Implements

IConnection.DefaultValueEncoding [▶ 76]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.10 AdsConnection.Disposed Property

Gets a value indicating whether this AdsConnection [▶ 357] is disposed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Disposed { get; }
```

Property Value

Type: Boolean
true if disposed; otherwise, false.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.11 AdsConnection.Id Property

Gets the AdsConnection [▶ 357] identifier.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Id { get; }
```

Property Value

Type: Int32
The identifier.

Implements

IConnection.Id [▶ 76]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.12 AdsConnection.IsActive Property

Gets a value indicating whether communication is in active state

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsActive { get; }
```

Property Value

Type: Boolean
true if this instance is active; otherwise, false.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.13 AdsConnection.IsConnected Property

Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

Type: Boolean
true if this instance is connected; otherwise, false.

Implements

IConnection.IsConnected [▶ 76]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.14 AdsConnection.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsLocal { get; }
```

Property Value

Type: Boolean
true if this instance is local; otherwise, false.

Implements

IAdsConnection.IsLocal [▶ 782]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.15 AdsConnection.IsLost Property

Gets a value indicating whether the communication is in lost / open state

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsLost { get; }
```

Property Value

Type: Boolean
true if this instance is lost; otherwise, false.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.16 AdsConnection.IsReconnecting Property

Gets a value indicating whether communication is ready for reconnecting

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReconnecting { get; }
```

Property Value

Type: Boolean
true if this instance is reconnecting; otherwise, false.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.17 AdsConnection.Name Property

Gets the name of this AdsConnection [▶ 357].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.18 AdsConnection.ResurrectingTries Property

Gets the number of tries to resurrect the AdsConnection [▶ 357].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ResurrectingTries { get; }
```

Property Value

Type: Int32
The number of tried resurrections of the IConnection [▶ 74].

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.19 AdsConnection.Resurrections Property

Gets the number of succeeded connection resurrections.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Resurrections { get; }
```

Property Value

Type: Int32
The resurrection count.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.20 AdsConnection.Session Property

Gets the Session object of the AdsConnection [▶ 357] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]
The client.

Implements

IConnection.Session [▶ 77]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.21 AdsConnection.State Property

Gets the current ConnectionState [▶ 379]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ConnectionState State { get; }
```

Property Value

Type: ConnectionState [▶ 67]
The state.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.1.22 AdsConnection.Timeout Property

Gets the timeout (in milliseconds)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Timeout { get; set; }
```

Property Value

Type: Int32
The timeout.

Implements

IConnection.Timeout [▶ 77]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2 AdsConnection Methods

The AdsConnection [▶ 357] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 402]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 403]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 405]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 406]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 564] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 408]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 409]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

	Name	Description
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 411]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 412]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 413]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 415]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	CleanupSymbolTable [▶ 416]	Clears the internal symbol cache.
	Close [▶ 417]	Closes the AdsConnection [▶ 357]
	Connect [▶ 417]	(Re)Connects the IConnection [▶ 74] when disconnected.
	CreateSymbolLoader [▶ 417]	Creates a new instance of the Symbol loader [▶ 1383] with the specified mode.
	CreateVariableHandle [▶ 429]	Generates a unique handle for an ADS variable.
	CreateVariableHandleAsync [▶ 429]	Determines the Symbol handle by its instance path asynchronously.
	DeleteDeviceNotification [▶ 430]	Deletes an existing notification.
	DeleteDeviceNotificationAsync [▶ 431]	Deletes a registered notification asynchronously.
	DeleteVariableHandle [▶ 432]	Releases the handle of a ADS variable again.

	Name	Description
	DeleteVariableHandleAsync [▶ 432]	Releases the handle of a ADS variable again (asynchronously)
	Disconnect [▶ 434]	Disconnects this IConnection [▶ 74].
	Dispose [▶ 434]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	InvokeRpcMethod(String, String, .Object.) [▶ 435]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 437]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 438]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 441]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 443]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 445]	Invokes the specified RPC Method asynchronously

	Name	Description
	Read(UInt32, Memory) [▶ 447]	
	Read(UInt32, UInt32, Memory) [▶ 448]	
	Read(UInt32, UInt32, Memory, Void) [▶ 448]	
	ReadAny(UInt32, Type) [▶ 451]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 453]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 455]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 456]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32., Int32) [▶ 457]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 452]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 454]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 461]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 463]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 465]	Reads data asynchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 466]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 459]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 460]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 462]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 464]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyString(UInt32, Int32, Encoding) [▶ 467]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 468]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 470]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 471]	read any string as an asynchronous operation.
	ReadAsync(UInt32, Memory, Void) [▶ 472]	
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 473]	
	ReadDataType [▶ 473]	Call this method to obtain information about the specified data type.

	Name	Description
	ReadDataTypeAsync [▶ 474]	read data type as an asynchronous operation.
	ReadDeviceInfo [▶ 475]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 476]	Reads the identification and version number of an ADS server.
	ReadState. [▶ 477]	Reads the ADS status and the device status from an ADS server.
	ReadState(Int32) [▶ 477]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 478]	Read the ADS State asynchronously
	ReadSymbol [▶ 479]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadSymbolAsync [▶ 479]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 481]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValue(String, Type) [▶ 483]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(String) [▶ 481]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 482]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 485]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync(String, Type, CancellationToken) [▶ 487]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(String, CancellationToken) [▶ 484]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 486]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadWrite(UInt32, Memory, Void) [▶ 488]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 489]	

	Name	Description
	ReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 489]	
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 491]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 491]	
	RegisterAdsStateChangedAsync [▶ 492]	Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.
	RegisterSymbolVersionChangedAsync [▶ 493]	Registers the symbol version changed asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 494]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 495]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 497]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 498]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	TryCreateVariableHandle [▶ 499]	Read (determine) the Symbol handle by its name/path

	Name	Description
	TryDeleteDeviceNotification [▶ 500]	Deletes a registered notification.
	TryDeleteVariableHandle [▶ 501]	Releases the specified symbol/variable handle synchronously.
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 503]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 505]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 506]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 508]	Invokes the rpc method.
	TryRead(UInt32, Memory, Void) [▶ 511]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 511]	
	TryRead(UInt32, UInt32, Memory, Void, Byte) [▶ 512]	
	TryReadDataType [▶ 513]	Call this method to obtain information about the specified data type.
	TryReadState(StateInfo.) [▶ 514]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

	Name	Description
	TryReadState(Int32, StateInfo.) [▶ 515]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadSymbol [▶ 515]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	TryReadValue(ISymbol, Object.) [▶ 517]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	TryReadValue(String, Type, Object.) [▶ 519]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 517]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 518]	Reads the value of a symbol and returns it as an object.
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 521]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 522]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte, ReadOnlyMemory) [▶ 522]	
	TryWrite(UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 525]	
	TryWriteControl(StateInfo) [▶ 526]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, Int32) [▶ 527]	Changes the ADS status and the device status of an ADS server.

	Name	Description
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 528]	
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 528]	
	TryWriteValue(String, Object) [▶ 529]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 531]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 530]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 532]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	UnregisterAdsStateChangedAsync [▶ 533]	Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.
	UnregisterSymbolVersionChangedAsync [▶ 533]	Unregisters the symbol version changed asynchronous.
	Write(UInt32, ReadOnlyMemory) [▶ 535]	
	Write(UInt32, UInt32) [▶ 535]	Trigger Client Method/Command.
	Write(UInt32, UInt32, Int32) [▶ 536]	Trigger Client Method/Command.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 537]	
	Write(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 537]	
	WriteAny(UInt32, Object) [▶ 538]	Writes an object synchronously to an ADS device.

	Name	Description
	WriteAny(UInt32, Object, .Int32.) [▶ 539]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 540]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 540]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object, .Int32., Int32) [▶ 541]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 542]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 543]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 545]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 545]	write any as an asynchronous operation.
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 547]	write any string as an asynchronous operation.
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 548]	write any string as an asynchronous operation.
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 549]	

	Name	Description
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 550]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 551]	
	WriteControl(StateInfo) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, Int32) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 553]	
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 554]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 555]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 555]	
	WriteSymbolAsync [▶ 556]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(String, Object) [▶ 557]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 559]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(String, T) [▶ 558]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue.T.(ISymbol, T) [▶ 560]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

	Name	Description
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 562]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 561]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 563]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.1 AdsConnection.AddDeviceNotification Method

Overload List

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 402]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 403]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotification(
    string variableName,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

variableName	Type: System.String Name of the variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotification Overload [▶ 402]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsConnection.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value.
Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotification Overload [▶ 402]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.AdsNotificationError [▶ 565]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.5.2.2 AdsConnection.AddDeviceNotificationAsync Method

Overload List

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 405]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 406]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 564] event.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AddDeviceNotificationAsync Method (String, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationAsync(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.

userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always be called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationAsync Overload [▶ 405]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsConnection.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 564] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431] should always be called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationAsync Overload [▶ 405]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

6.2.5.2.3 AdsConnection.AddDeviceNotificationEx Method

Overload List

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 408]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 409]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 411]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 412]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationEx Overload [▶ 408]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsConnection.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationEx Overload [▶ 408]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsConnection.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationEx Overload [▶ 408]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.AdsNotificationError [▶ 565]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

AdsConnection.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument.
args	Type: .System.Int32. Additional arguments for 'AnyType' types.

Return Value

Type: UInt32
The notification handle.

Implements

IAdsNotifications.AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationEx Overload [▶ 408]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotificationEx [▶ 566]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

6.2.5.2.4 AdsConnection.AddDeviceNotificationExAsync Method

Overload List

	Name	Description
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 413]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 415]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431] should always be called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationExAsync Overload [▶ 413]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

AdsConnection.AddDeviceNotificationExAsync Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle) and the ErrorCode [▶ 992] after execution.

Implements

IAdsNotifications.AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

AddDeviceNotificationExAsync Overload [▶ 413]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 431]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.AdsNotificationError [▶ 565]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.5.2.5 AdsConnection.CleanupSymbolTable Method

Clears the internal symbol cache.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CleanupSymbolTable()
```

Implements

IAdsSymbolicAccess.CleanupSymbolTable. [▶ 942]

Remarks

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.6 AdsConnection.Close Method

Closes the AdsConnection [▶ 357]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Close()
```

Implements

IConnection.Close. [▶ 78]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.7 AdsConnection.Connect Method

(Re)Connects the IConnection [▶ 74] when disconnected.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Connect()
```

Return Value

Type: Boolean
true if the AdsConnection [▶ 357] is reconnected, false otherwise.

Implements

IConnection.Connect. [▶ 78]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.8 AdsConnection.CreateSymbolLoader Method

Creates a new instance of the Symbol loader [▶ 1383] with the specified mode.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IAdsSymbolLoader CreateSymbolLoader(
    ISession session,
    ISymbolLoaderSettings settings
)
```

Parameters

session	Type: TwinCAT.ISession [▶ 88] The session (for session orientated loads/symbols). Can be NULL if not present.
settings	Type: TwinCAT.ISymbolLoaderSettings [▶ 99] The settings.

Return Value

Type: IAdsSymbolLoader [▶ 1383]
The IAdsSymbolLoader [▶ 1383] interface of the Symbol loader.

Exceptions

Exception	Condition
ObjectDisposedException	
ObjectDisposedException	

Remarks

The Symbol Loader (V2) supports the following modes [▶ 149]. Flat [▶ 149]The flat mode organizes the Symbols in a flat list. At the beginning this List caches only the root symbol objects, which can be enumerated. To access the sub elements like structure fields or array elements use the SubSymbols [▶ 2182] collection. The property get accessor generates the subsymbols lazy on the fly (performance optimized) and stores them internally as weak reference (memory optimized). This mode is available in all .NET versions.VirtualTree [▶ 149]On top of the behaviour of the Flat [▶ 149], the virtual tree mode organizes the Symbols hierarchically with parent-child relationships. That eases the access to the hierarchical structure but needs slightly more preprocessing of the data. This mode is available in all .NET Versions. DynamicTree [▶ 149]The Dynamic tree mode organizes the Symbols hierarchically and (dynamically) creates struct members, array elements and enum fields on the fly. 'Dynamically' means here not only lazy creation like in Flat [▶ 149], but furthermore real creation of type safe .NET complex types/instances as represetantives of the TwinCAT Symbol objects/types. This feature is only available on platforms that support the Dynamic Language Runtime (DLR); actually all .NET Framework Version larger than 4.0. Virtual instances means, that all Symbols are ordered within a tree structure. For that symbol nodes that are not located on a fixed address, a Virtual Symbol will be created. Setting the virtualInstance parameter to 'false' means, that the located symbols will be returned in a flattened list.

Examples

The following sample shows how to create a dynamic version of the SymbolLoader V2. The dynamic symbol loader makes use of the Dynamic Language Runtime (DLR) of the .NET Framework. That means Structures, Arrays and Enumeration types and instances are generated 'on-the-fly' during symbol Browsing. These created dynamic objects are a one to one representation of the Symbol Server target objects (e.g the IEC61131 types on the PLC). Dynamic language features are only available from .NET4 upwards.

Dynamic Tree Mode

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;

        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;
```

```
        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
        }
    }

    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
```

```
f"));
        }
    }
    }
}
```

The following sample shows how to create a static (non dynamic) version of the SymbolLoader V2. The static symbol loader in version 2 is a nearly code compatible version of the Dynamic Loader, only the dynamic creation of objects is not available. The reason for supporting this mode is that .NET Framework Versions lower than Version 4.0 (CLR2) doesn't support the Dynamic Language Runtime (DLR). The SymbolLoader V2 static object is supported from .NET 2.0 on.

Virtual Tree Mode

```
using System;
using System.Threading;
using System.Diagnostics;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.Ads.TypeSystem;

namespace Sample
{
    class SymbolBrowserProgramV2VirtualTree
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        ConsoleLogger logger = new ConsoleLogger();

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for start:");
        Console.ReadLine();

        //logger.Active = false;

        Stopwatch stopper = new Stopwatch();

        // Parse the command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        stopper.Start();

        using (AdsClient client = new AdsClient())
        {
        //client.Synchronize = false;

        // Connect the AdsClient to the device target.
        client.Connect(address);

        // Creates the Symbol Objects as hierarchical tree
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree, ValueA
ccessMode.IndexGroupOffsetPreferred);
        ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

        // Dump Datatypes from Target Device
        Console.WriteLine(string.Format("Dumping '{0}' DataTypes:", symbolLoader.DataTypes.Count));
        foreach (IDataType type in symbolLoader.DataTypes)
        {
            logger.DumpType(type);
        }
        Console.WriteLine("");

        // Dump Symbols from target device
        Console.WriteLine("Dumping '{0}' Symbols:", symbolLoader.Symbols.Count);
        foreach (ISymbol symbol in symbolLoader.Symbols)
        {
            logger.DumpSymbol(symbol,0);
        }
        }
        stopper.Stop();
        TimeSpan elapsed = stopper.Elapsed;
```

```
        Console.WriteLine("");
        Console.WriteLine("Browsing complete tree: {0},
({1} DataTypes, {2} Symbols)",elapsed,logger.DataTypesCount,logger.DataTypesCount);
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }
```

Examples

The SymbolLoader V2 static object is supported from .NET 2.0 on.

Flat Mode

```
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SymbolBrowserProgramV2Flat
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        ConsoleLogger logger = new ConsoleLogger();

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for start:");
        Console.ReadLine();

        //logger.Active = false;

        Stopwatch stopper = new Stopwatch();

        // Parse the command line arguments
        AmsAddress address = ArgParser.Parse(args);

        stopper.Start();

        // Create the ADS Client
        using (AdsClient client = new AdsClient())
        {
        //client.Synchronize = false;

        // Connect to Address
        client.Timeout = 30000;
        client.Connect(address);

        // Creates the Symbol Objects in Flat Mode (Flat list)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.Flat, ValueAccessMo
de.IndexGroupOffsetPreferred);
        ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

        // Dump Datatypes from Target Device
        Console.WriteLine(string.Format("Dumping '{0}' DataTypes:",symbolLoader.DataTypes.Count));
        foreach (IDataType type in symbolLoader.DataTypes)
        {
            logger.DumpType(type);
        }

        Console.WriteLine("");

        // Dump Symbols from target device
        Console.WriteLine("Dumping '{0}' Symbols:",symbolLoader.Symbols.Count);
        foreach (ISymbol symbol in symbolLoader.Symbols)
        {
            logger.DumpSymbol(symbol,0);
        }
```

```
        }
        stopper.Stop();
        TimeSpan elapsed = stopper.Elapsed;

        Console.WriteLine("");
        Console.WriteLine("Browsing complete tree: {0},
({1} DataTypes, {2} Symbols)", elapsed, logger.DataTypesCount, logger.DataTypesCount);
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }
```

Examples

Argument Parser

```
public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
    AmsNetId netId = AmsNetId.Local;
    int port = 851;

    if (args != null)
    {
        if (args.Length > 0 && args[0] != null)
        netId = AmsNetId.Parse(args[0]);

        if (args.Length > 1 && args[1] != null)
        port = int.Parse(args[1]);
    }
    return new AmsAddress(netId, port);
    }
}
```

Dumping Symbols

```
/// <summary>
/// Console logger
/// </summary>
public class ConsoleLogger
{
    public ConsoleLogger()
    {
    }
    bool _active = true;

    /// <summary>
    /// Gets or sets a value indicating whether this ConsoleLogger is active.
    /// </summary>
    /// <value><c>true</c> if active; otherwise, <c>false</c>.</value>
    public bool Active
    {
    get { return _active; }
    set
    {
        _active = value;
    }
    }

    int _dataTypes = 0;

    /// <summary>
    /// Gets the number of dumped dataTypes.
    /// </summary>
    /// <value>The data types count.</value>
    public int DataTypesCount
    {
    get { return _dataTypes; }
    }

    int _symbols = 0;

    /// <summary>
```

```
    /// Gets the number of dumped symbols
    /// </summary>
    /// <value>The symbols count.</value>
    public int SymbolsCount
    {
    get { return _symbols; }
    }


    /// <summary>
    /// Dumps the data type.
    /// </summary>
    /// <param name="dataType">Data Type.</param>
    public void DumpType(IDataType dataType)
    {
    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType.Categ
ory, dataType.Size));

    switch (dataType.Category)
    {
        case DataTypeCategory.Alias:
        IAliasType alias = (IAliasType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Alias BaseType: {0}", alias.BaseTypeName));
        break;

        case DataTypeCategory.Enum:

        //IEnumType<ushort> enumType = (IEnumType<ushort>)dataType;
        IEnumType enumType = (IEnumType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Enum BaseType: {0}", enumType.BaseTypeName));

        foreach (IEnumValue enumValue in enumType.EnumValues)
        {
            WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumValu
e.Primitive));
        }
        break;
        case DataTypeCategory.Array:

        IArrayType arrayType = (IArrayType)dataType;
        int i = 0;

        foreach (IDimension dim in arrayType.Dimensions)
        {
            WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}", i+
+, dim.LowerBound, dim.ElementCount));
        }
        break;
        case DataTypeCategory.Struct:
        IStructType structType = (IStructType)dataType;

        foreach (IMember member in structType.Members)
        {
            WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}", member.Offset
, member.InstanceName, member.TypeName));
        }
        break;
        default:
        break;
    }

    foreach (ITypeAttribute attribute in dataType.Attributes)
    {
        WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
    }
    if (!string.IsNullOrEmpty(dataType.Comment))
    {
        WriteLine(GetPrefix(1) + string.Format("Comment: {0}", dataType.Comment));
    }

    IRpcCallableType rpcCallable = dataType as IRpcCallableType;

    if (rpcCallable != null)
    {
        foreach (IRpcMethod rpcMethod in rpcCallable.RpcMethods)
        {
        if (string.IsNullOrEmpty(rpcMethod.Comment))
            WriteLine(GetPrefix(1) + string.Format("Method: {0}", rpcMethod));
        else
            WriteLine(GetPrefix(1) + string.Format("Method: {0}, Comment: {1}", rpcMethod, rpcMethod
```

```
.Comment));
        }
    }
    _dataTypes++;
    }

    ///// <summary>
    ///// Dumps the Datatype to Console
    ///// </summary>
    ///// <param name="dataType">DataType.</param>
    //public void DumpType(ITcAdsDataType dataType)
    //{
    //    // Dump the Attributes (PLC Metadata)
    //    foreach (ITypeAttribute attribute in dataType.Attributes)
    //    {
    //    WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value)
);
    //    }

    //    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType
.Category, dataType.Size));

    //    if (dataType.BaseType != null)
    //    {
    //    WriteLine(GetPrefix(1) + string.Format("BaseType: {0}", dataType.BaseType));
    //    }

    //    switch (dataType.Category)
    //    {
    //    case DataTypeCategory.Enum:
    //        foreach (IEnumValue enumValue in dataType.EnumValues)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumVa
lue.Primitive));
    //        }
    //        break;
    //    case DataTypeCategory.Array:
    //        int i = 0;
    //        foreach (IDimension dim in dataType.Dimensions)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}", i+
+, dim.LowerBound, dim.ElementCount));
    //        }
    //        break;
    //    case DataTypeCategory.Struct:
    //        foreach (ITcAdsSubItem subItem in dataType.SubItems)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}", subItem.Off
set, subItem.SubItemName, subItem.Name));
    //        }
    //        break;
    //    default:
    //        break;
    //    }
    //    _dataTypes++;
    //}

    /// <summary>
    /// Dump Symbol
    /// </summary>
    /// <param name="symbol">The symbol.</param>
    /// <param name="level">Output indentation level</param>
    public void DumpSymbol(ISymbol symbol, int level)
    {
    IDataType type = symbol.DataType as IDataType;

    foreach (ITypeAttribute attribute in symbol.Attributes)
    {
        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Valu
e));
    }

    WriteLine(GetPrefix(level) + string.Format("{0} : {1} (IG: 0x{2} IO: 0x{3} size:
{4})", symbol.InstanceName, symbol.TypeName, ((IAdsSymbol)symbol).IndexGroup.ToString("x"), ((IAdsSy
mbol)symbol).IndexOffset.ToString("x"), symbol.Size));

    if (symbol.Category == DataTypeCategory.Array)
    {
        IArrayInstance arrInstance = (IArrayInstance)symbol;
        IArrayType arrType = (IArrayType)symbol.DataType;
```

```
        int count = 0;
        level++;

        foreach (ISymbol arrayElement in arrInstance.Elements)
        {
        DumpSymbol(arrayElement, level);
        count++;

        if (count > 20) // Write only the first 20 to limit output
            break;
        }
    }
    else if (symbol.Category == DataTypeCategory.Struct)
    {
        IStructInstance structInstance = (IStructInstance)symbol;
        IStructType structType = (IStructType)symbol.DataType;

        level++;

        foreach (ISymbol member in structInstance.MemberInstances)
        {
        DumpSymbol(member, level);
        }
    }
    _symbols++;
    }

    ///// <summary>
    ///// Dumps the specified Symbol to the Console
    ///// </summary>
    ///// <param name="symbol">The symbol.</param>
    ///// <param name="level">The level.</param>
    //public void DumpSymbol(IAdsSymbol2 symbol, int level)
    //{
    //    // Dump Attributes of the Symbol
    //    foreach (ITypeAttribute attribute in symbol.Attributes)
    //    {
    //    WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Va
lue));
    //    }

    //    ITcAdsSymbolBrowser subSymbolProvider = (ITcAdsSymbolBrowser)symbol;

    //    // Dump The Symbol
    //    WriteLine(GetPrefix(level) + string.Format("{0} : {1} ({2}, IG: 0x{3} IO: 0x{4} size:
{6} subCount:
{5})", symbol.Name, symbol.TypeName, symbol.DataTypeId, symbol.IndexGroup.ToString("x"), symbol.Inde
xOffset.ToString("x"), subSymbolProvider.SubSymbols.Count, symbol.Size));
    //    level++;

    //    // Dump all SubSymbols with indentation
    //    foreach (IAdsSymbol2 subSymbol in ((ITcAdsSymbolBrowser)symbol).SubSymbols)
    //    {
    //    DumpSymbol(subSymbol, level);
    //    }
    //    _symbols++;
    //}

    /// <summary>
    /// Dump namespace.
    /// </summary>
    /// <param name="ns">The namespace.</param>
    public void DumpNamespace(INamespace<IDataType> ns)
    {
    WriteLine("Namespace: {0}, DataTypes: {1}", ns.Name, ns.DataTypes.Count);

    foreach (IDataType type in ns.DataTypes)
    {
        DumpType(type);
    }
    }

    /// <summary>
    /// Get the indentation prefix
    /// </summary>
    /// <param name="level">The level.</param>
    /// <returns>System.String.</returns>
    public string GetPrefix(int level)
    {
```

```
    return "".PadLeft(level * 3);
    }

    /// <summary>
    /// Writes a line to the Console
    /// </summary>
    /// <param name="message">The message.</param>
    public void WriteLine(string message)
    {
    if (Active)
    {
        Console.WriteLine(message);
    }
    }

    /// <summary>
    /// Writes a line to the console
    /// </summary>
    /// <param name="format">The format.</param>
    /// <param name="args">The arguments.</param>
    public void WriteLine(string format, params object[] args)
    {
    if (Active)
    {
        Console.WriteLine(format, args);
    }
    }
}
```

Examples

The following sample shows how to call (Remote Procedures / Methods) with Virtual Symbols

RPC Call in Virtual Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
```

```
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Examples

The following sample shows how to call (Remote Procedures / Methods) with Dynamic Symbols.

RPC Call in Dynamic Mode

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;

    class RpcCallDynamicProgram
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);
        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        ISymbolLoader dynLoader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)
        ResultDynamicSymbols resultGetSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSy
mbolsAsync(cancel);
        dynamic symbols = resultGetSymbols.Symbols;
        dynamic main = symbols.Main; // Gets the MAIN Instance of the PLC Program

         // Call a Method that has the following signature (within MAIN Program)

        /*  {attribute 'TcRpcEnable'}
            METHOD PUBLIC M_Add : INT
            VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
            END_VAR
        */

        short result = main.M_Add(3,4); // Synchronous Call

        // Call a Method that has no parameter and returns VOID
        main.M_Method1(); // Synchronous call

        //Browsing Rpc Methods
        foreach (IRpcMethod method in main.RpcMethods)
        {
            string methodName = method.Name;

            foreach (IRpcMethodParameter parameter in method.Parameters)
```

```
            {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
            }
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.TypeSystem.SymbolLoaderFactory [▶ 1523]

6.2.5.2.9 AdsConnection.CreateVariableHandle Method

Generates a unique handle for an ADS variable.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint CreateVariableHandle(
    string variableName
)
```

Parameters

variableName	Type: System.String Name of the ADS variable

Return Value

Type: UInt32
The handle of the ADS Variable.

Implements

IAdsHandle.CreateVariableHandle(String) [▶ 829]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.10 AdsConnection.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultHandle> CreateVariableHandleAsync(
    string variableName,
    CancellationToken cancel
)
```

Parameters

variableName	Type: System.String Name of the variable.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'CreateVariableHandle' operation. The ResultHandle [▶ 1005] parameter contains the variable handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandleAsync(String, CancellationToken) is the DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 432]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 432]

AdsConnection.TryCreateVariableHandle(String, UInt32.) [▶ 499]

AdsConnection.CreateVariableHandle(String) [▶ 429]

6.2.5.2.11 AdsConnection.DeleteDeviceNotification Method

Deletes an existing notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void DeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Handle of the notification.

Implements

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.12 AdsConnection.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'DeleteDeviceNotification' operation. The ErrorCode [▶ 992] property contains the ADS error code after execution.

Implements

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

Remarks

This is the complementary method to AddDeviceNotificationAsync Overload [▶ 847] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotificationAsync Overload [▶ 847]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotification Overload [▶ 844]

6.2.5.2.13 AdsConnection.DeleteVariableHandle Method

Releases the handle of a ADS variable again.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void DeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Implements

IAdsHandle.DeleteVariableHandle(UInt32) [▶ 830]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.14 AdsConnection.DeleteVariableHandleAsync Method

Releases the handle of a ADS variable again (asynchronously)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'ReadState' operation. The ResultAds [▶ 989] parameter contains the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831] is the CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.CreateVariableHandleAsync(String, CancellationToken) [▶ 429]

AdsConnection.TryDeleteVariableHandle(UInt32) [▶ 501]

AdsConnection.DeleteVariableHandle(UInt32) [▶ 432]

6.2.5.2.15 AdsConnection.Disconnect Method

Disconnects this IConnection [▶ 74].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Disconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IConnection.Disconnect. [▶ 79]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.16 AdsConnection.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.17 AdsConnection.InvokeRpcMethod Method

Overload List

	Name	Description
	InvokeRpcMethod(String, String, .Object.) [▶ 435]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 437]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 438]	Invokes the specified RPC Method

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethod Method (String, String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object.) [▶ 890]

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
```

```
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethod Overload [▶ 435]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethod Method (String, String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL
outParameters	Type: .System.Object.. The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethod Overload [▶ 435]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifiying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifiying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IAdsRpcInvoke.InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
```

```
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethod Overload [▶ 435]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.18 AdsConnection.InvokeRpcMethodAsync Method

Overload List

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 441]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 443]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 445]	Invokes the specified RPC Method asynchronously

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
```

```
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethodAsync Overload [▶ 440]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [▶ 1028] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethodAsync Overload [▶ 440]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The RPC callable symbol."
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [▶ 1028] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsRpcInvoke.InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
```

```
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

InvokeRpcMethodAsync Overload [▶ 440]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.19 AdsConnection.Read Method

Overload List

	Name	Description
	Read(UInt32, Memory) [▶ 447]	
	Read(UInt32, UInt32, Memory) [▶ 448]	
	Read(UInt32, UInt32, Memory, Void) [▶ 448]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Read Method (UInt32, Memory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Read(
    uint variableHandle,
    Memory buffer
)
```

Parameters

variableHandle	Type: System.UInt32
buffer	Type: Memory

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

Read Overload [▶ 447]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Read Method (UInt32, UInt32, Memory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

Read Overload [▶ 447]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Read Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
timeout	Type: System.Void

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

Read Overload [▶ 447]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.20 AdsConnection.ReadAny Method

Overload List

	Name	Description
	ReadAny.T.(UInt32) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T. (UInt32, .Int32.) [▶ 450]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type) [▶ 451]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 452]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 453]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 454]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 455]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 456]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32., Int32) [▶ 457]	Reads data synchronously from an ADS device and writes it to an object.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny.T. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32) [▶ 707]

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny.T. Method (UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, .Int32.) [▶ 708]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny Method (UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint variableHandle,
    Type type
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read object.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type) [▶ 709]

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny.T. Method (UInt32, UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, UInt32) [▶ 710]

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny Method (UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, Type, .Int32.) [▶ 710]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny.T. Method (UInt32, UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Implements

IAdsAnyAccess.ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny Method (UInt32, UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type) [▶ 712]

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny Method (UInt32, UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Implements

IAdsAnyAccess.ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAny Method (UInt32, UInt32, Type, .Int32., Int32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    int timeout
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
timeout	Type: System.Int32 The timeout.

Return Value

Type: Object
The Value of the data marshalled to the specified type.

Implements

IAdsReadWriteTimeoutAccess.ReadAny(UInt32, UInt32, Type, .Int32., Int32) [▶ 880]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args. The type is limited to Primitive types ('AnyType').

Reference

AdsConnection Class [▶ 357]

ReadAny Overload [▶ 449]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.21 AdsConnection.ReadAnyAsync Method

Overload List

	Name	Description
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 459]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 460]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 461]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 462]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 463]	Reads data synchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 464]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 465]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 466]	Reads data asynchronously from an ADS device and writes it to an object.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync.T. Method (UInt32, CancellationToken)

Reads data synchronously from an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable/symbol handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The Type of the value to be read.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]

Remarks

As object types only primitive types are supported.

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync.T. Method (UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	Type of the object to be read

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync Method (UInt32, Type, CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable/symbol handle.
type	Type: System.Type Type of the object to be read.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]

Remarks

As object types only primitive types are supported.

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync.T. Method (UInt32, UInt32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T

Return Value

Type: Task.ResultValue [▶ 1029].T..
The asynchronous result.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].

Type of value Parameter	Necessary Arguments (args)
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync.T. Method (UInt32, UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The type of the result value.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

ReadAnyAsync Overload [▶ 458]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.22 AdsConnection.ReadAnyString Method

Overload List

	Name	Description
	ReadAnyString(UInt32, Int32, Encoding) [▶ 467]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 468]	Reads as string from a specified address.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyString Method (UInt32, Int32, Encoding)

Reads a string from the specified symbol/variable.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
The string value.

Implements

IAdsAnyAccess.ReadAnyString(UInt32, Int32, Encoding) [▶ 722]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

ReadAnyString Overload [▶ 467]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)

Reads as string from a specified address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.

indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
System.String.

Implements

IAdsAnyAccess.ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

ReadAnyString Overload [▶ 467]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.23 AdsConnection.ReadAnyStringAsync Method

Overload List

	Name	Description
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 470]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 471]	read any string as an asynchronous operation.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read string (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

ReadAnyStringAsync Overload [▶ 469]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

read any string as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value) and the ErrorCode [▶ 992] after execution.

Implements

IAdsAnyAccess.ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

ReadAnyStringAsync Overload [▶ 469]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.24 AdsConnection.ReadAsync Method

Overload List

	Name	Description
	ReadAsync(UInt32, Memory, Void) [▶ 472]	
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 473]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAsync Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

AdsConnection Class [▶ 357]

ReadAsync Overload [▶ 472]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadAsync Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

AdsConnection Class [▶ 357]

ReadAsync Overload [▶ 472]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.25 AdsConnection.ReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ReadDataType(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the data type (without namespace)

Return Value

Type: IDataType [▶ 1986]
An containing the requested type.

Implements

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [▶ 943]

6.2.5.2.26 AdsConnection.ReadDataTypeAsync Method

read data type as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```

Parameters

typeName	Type: System.String Name of the data type.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IDataType [▶ 1986]..
A task that represents the asynchronous 'ReadDataType' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [▶ 943]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

6.2.5.2.27 AdsConnection.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DeviceInfo ReadDeviceInfo()
```

Return Value

Type: DeviceInfo [▶ 698]
DeviceInfo struct containing the name of the device and the version information.

Implements

IAdsConnection.ReadDeviceInfo. [▶ 795]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.28 AdsConnection.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultDeviceInfo [▶ 1001].
A task that represents the asynchronous 'ReadDeviceState' operation. The ResultDeviceInfo [▶ 1001] parameter contains the value DeviceInfo [▶ 1003] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsConnection.ReadDeviceInfoAsync(CancellationToken) [▶ 796]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.29 AdsConnection.ReadState Method

Overload List

	Name	Description
	ReadState. [▶ 477]	Reads the ADS status and the device status from an ADS server.
	ReadState(Int32) [▶ 477]	Reads the ADS status and the device status from an ADS server.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadState Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo ReadState()
```

Return Value

Type: StateInfo [▶ 1041]
The ADS statue and device status.

Implements

IAdsStateProvider.ReadState. [▶ 931]

Reference

AdsConnection Class [▶ 357]

ReadState Overload [▶ 477]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadState Method (Int32)

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo ReadState(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout.

Return Value

Type: StateInfo [▶ 1041]
The ADS statue and device status.

Implements

IAdsStateControlTimeout.ReadState(Int32) [▶ 923]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

ReadState Overload [▶ 477]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.30 AdsConnection.ReadStateAsync Method

Read the ADS State asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadDeviceState [▶ 1016].
A task that represents the asynchronous 'ReadState' operation. The ResultReadDeviceState [▶ 1016] parameter contains the value State [▶ 1018] and the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.ReadStateAsync(CancellationToken) [▶ 931]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.31 AdsConnection.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IAdsSymbol ReadSymbol(
    string name
)
```

Parameters

name	Type: System.String Name of the symbol.

Return Value

Type: IAdsSymbol [▶ 1379]
A IAdsSymbol [▶ 1379] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.ReadSymbol(String) [▶ 943]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.32 AdsConnection.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(
    string name,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IAdsSymbol [▶ 1379]..
A task that represents the asynchronous 'ReadSymbolInfo' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken) [▶ 944]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.33 AdsConnection.ReadValue Method

Overload List

	Name	Description
	ReadValue.T.(String) [▶ 481]	Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.
	ReadValue(ISymbol) [▶ 481]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValue.T.(ISymbol) [▶ 482]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 483]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValue.T. Method (String)

Reads the value of a symbol and returns the value. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadValue<T>(
    string name
)
```

Parameters

name	Type: System.String Name of the ADS symbol.

Type Parameters

T	The value type

Return Value

Type: T
Value of the symbol

Implements

IAdsSymbolicAccess.ReadValue.T.(String) [▶ 945]

Reference

AdsConnection Class [▶ 357]

ReadValue Overload [▶ 480]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValue Method (ISymbol)

Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes(UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Return Value

Type: Object
The value of the symbol as an object.

Implements

IAdsSymbolicAccess.ReadValue(ISymbol) [▶ 946]

Reference

AdsConnection Class [▶ 357]

ReadValue Overload [▶ 480]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValue.T. Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Type Parameters

T	The value type.

Return Value

Type: T
The value of the symbol.

Implements

IAdsSymbolicAccess.ReadValue.T.(ISymbol) [▶ 946]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsConnection Class [▶ 357]

ReadValue Overload [▶ 480]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValue Method (String, Type)

Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    string name,
    Type type
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.

Return Value

Type: Object
Value of the symbol

Implements

IAdsSymbolicAccess.ReadValue(String, Type) [▶ 947]

Reference

AdsConnection Class [▶ 357]

ReadValue Overload [▶ 480]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.34 AdsConnection.ReadValueAsync Method

Overload List

	Name	Description
	ReadValueAsync.T. (String, CancellationToken) [▶ 484]	Reads the value of a symbol asynchronously.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 485]	Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.
	ReadValueAsync.T. (ISymbol, CancellationToken) [▶ 486]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync(String, Type, CancellationToken) [▶ 487]	Reads the value of a symbol asynchronously.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValueAsync.T. Method (String, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadValueAsync<T>(
    string name,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync.T.(String, CancellationToken) [▶ 948]

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [▶ 357]

ReadValueAsync Overload [▶ 484]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValueAsync Method (ISymbol, CancellationToken)

Reads the value of a symbol and returns it as an object. Strings and all primitive datatypes (UInt32, Int32, Bool etc.) are supported. Arrays and structures cannot be read.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
The value of the symbol as an object.

Implements

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [▶ 949]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

ReadValueAsync Overload [▶ 484]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValueAsync.T. Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultValue<T>> ReadValueAsync<T>(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 950]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsConnection Class [▶ 357]

ReadValueAsync Overload [▶ 484]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAnyValue> ReadValueAsync(
    string name,
    Type type,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [▶ 951]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [▶ 357]

ReadValueAsync Overload [▶ 484]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.35 AdsConnection.ReadWrite Method

Overload List

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 488]	
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 489]	
	ReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 489]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadWrite Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

ReadWrite Overload [▶ 488]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadWrite Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

ReadWrite Overload [▶ 488]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
timeout	Type: System.Byte

Return Value

Type: Int32

Reference

AdsConnection Class [▶ 357]

ReadWrite Overload [▶ 488]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.36 AdsConnection.ReadWriteAsync Method

Overload List

	Name	Description
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 491]	
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 491]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadWriteAsync Method (UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

AdsConnection Class [▶ 357]

ReadWriteAsync Overload [▶ 490]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ReadWriteAsync Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

AdsConnection Class [▶ 357]

ReadWriteAsync Overload [▶ 490]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.37 AdsConnection.RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be registered for AdsStateChanged calls.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.RegisterAdsStateChangedAsync(EventHandler.AdsStateChangedEventArgs., CancellationToken) [▶ 932]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.38 AdsConnection.RegisterSymbolVersionChangedAsync Method

Registers the symbol version changed asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to register.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.RegisterSymbolVersionChangedAsync(EventHandler.AdsSymbolVersionChangedEventArgs., CancellationToken) [▶ 935]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.39 AdsConnection.TryAddDeviceNotification Method

Overload List

	Name	Description
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 494]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 495]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotification(
    string variableName,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

variableName	Type: System.String Name of the variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data.

handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS ErrorCode.

Implements

IAdsNotifications.TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 500] should always be called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

TryAddDeviceNotification Overload [▶ 494]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.TryDeleteDeviceNotification(UInt32) [▶ 500]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

AdsConnection.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 564] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsNotifications.TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 564] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 864] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

TryAddDeviceNotification Overload [▶ 494]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryDeleteDeviceNotification(UInt32) [▶ 500]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.AdsNotificationError [▶ 565]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

[Overload:TwinCAT.Ads.IAdsNotifications..AddDeviceNotificationAsync] Overload

6.2.5.2.40 AdsConnection.TryAddDeviceNotificationEx Method

Overload List

	Name	Description
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 497]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 498]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

Parameters

symbolPath	Type: System.String The symbol path..

settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument.
args	Type: .System.Int32. Additional arguments.
handle	Type: System.UInt32. The handle.

Return Value

Type: AdsErrorCode [▶ 575]
The handle of the notification.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TryAddDeviceNotificationEx Overload [▶ 497]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryAddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 566] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. The 'AnyType' arguments.
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error code.

Implements

IAdsNotifications.TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 430] should always called when the notification is not used anymore.

Reference

AdsConnection Class [▶ 357]

TryAddDeviceNotificationEx Overload [▶ 497]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.DeleteDeviceNotification(UInt32) [▶ 430]

AdsConnection.AdsNotificationEx [▶ 566]

AdsConnection.AdsNotificationError [▶ 565]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.5.2.41 AdsConnection.TryCreateVariableHandle Method

Read (determine) the Symbol handle by its name/path

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryCreateVariableHandle(
    string symbolName,
    out uint variableHandle
)
```

Parameters

symbolName	Type: System.String SymbolName / Path.
variableHandle	Type: System.UInt32. The handle.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [▶ 834]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.42 AdsConnection.TryDeleteDeviceNotification Method

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [▶ 864]

Remarks

This is the complementary method to TryAddDeviceNotification Overload [▶ 859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

AdsConnection.AdsNotification [▶ 564]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.5.2.43 AdsConnection.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Implements

IAdsHandle.TryDeleteVariableHandle(UInt32) [▶ 835]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32.) [▶ 499]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryCreateVariableHandle(String, UInt32.) [▶ 499]

AdsConnection.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 432]

AdsConnection.DeleteVariableHandle(UInt32) [▶ 432]

6.2.5.2.44 AdsConnection.TryInvokeRpcMethod Method

Overload List

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 503]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 505]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 506]	Invokes the rpc method.

	Name	Description
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 508]	Invokes the rpc method.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryInvokeRpcMethod Method (String, String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

TryInvokeRpcMethod Overload [▶ 502]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryInvokeRpcMethod Method (String, String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
```

```
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

TryInvokeRpcMethod Overload [▶ 502]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
```

```
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol.
methodName	Type: System.String Name of the method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);
```

```
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

TryInvokeRpcMethod Overload [▶ 502]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryInvokeRpcMethod Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryInvokeRpcMethod(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The RPC callable symbol
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsRpcInvoke.TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
```

```
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

AdsConnection Class [▶ 357]

TryInvokeRpcMethod Overload [▶ 502]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.45 AdsConnection.TryRead Method

Overload List

	Name	Description
	TryRead(UInt32, Memory, Void) [▶ 511]	
	TryRead(UInt32, UInt32, Memory, Void) [▶ 511]	
	TryRead(UInt32, UInt32, Memory, Void, Byte) [▶ 512]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryRead Method (UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryRead Overload [▶ 510]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryRead Method (UInt32, UInt32, Memory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryRead Overload [▶ 510]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryRead Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout,
    byte readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
timeout	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryRead Overload [▶ 510]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.46 AdsConnection.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)
```

Parameters

typeName	Type: System.String Name of the symbol.
dataType	Type: TwinCAT.TypeSystem.IDataType [▶ 1986]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
A IDataType [▶ 1986] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	typeName
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

6.2.5.2.47 AdsConnection.TryReadState Method

Overload List

	Name	Description
	TryReadState(StateInfo.) [▶ 514]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryReadState(Int32, StateInfo.) [▶ 515]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadState Method (StateInfo.)

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode of the ads read state call. Check for AdsErrorCode.NoError to see if call was successful.

Implements

IAdsStateProvider.TryReadState(StateInfo.) [▶ 932]

Reference

AdsConnection Class [▶ 357]

TryReadState Overload [▶ 514]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadState Method (Int32, StateInfo.)

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadState(
    int timeout,
    out StateInfo stateInfo
)
```

Parameters

timeout	Type: System.Int32 The timeout.
stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode of the ads read state call. Check for AdsErrorCode.NoError to see if call was successful.

Implements

IAdsStateControlTimeout.TryReadState(Int32, StateInfo.) [▶ 924]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TryReadState Overload [▶ 514]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.48 AdsConnection.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadSymbol(
    string symbolPath,
    out IAdsSymbol symbol
)
```

Parameters

symbolPath	Type: System.String Name of the symbol.
symbol	Type: TwinCAT.Ads.TypeSystem.IAdsSymbol [▶ 1379]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
A IAdsSymbol [▶ 1379] containing the requested symbol information or null if symbol could not be found.

Implements

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol.) [▶ 953]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.49 AdsConnection.TryReadValue Method

Overload List

	Name	Description
	TryReadValue.T.(String, T.) [▶ 517]	Reads the value of a symbol and returns the value as object.
	TryReadValue(ISymbol, Object.) [▶ 517]	Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.
	TryReadValue.T.(ISymbol, T.) [▶ 518]	Reads the value of a symbol and returns it as an object.
	TryReadValue(String, Type, Object.) [▶ 519]	Reads the value of a symbol and returns the value as object.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadValue.T. Method (String, T.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue<T>(
    string name,
    out T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T. The read value of the Symbol.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Implements

IAdsSymbolicAccess.TryReadValue.T.(String, T.) [▶ 954]

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [▶ 357]

TryReadValue Overload [▶ 516]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadValue Method (ISymbol, Object.)

Reads the value of a symbol and returns the value as object. The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue(
    ISymbol symbol,
    out Object value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: System.Object. The value.

Return Value

Type: AdsErrorCode [▶ 575]
Value of the symbol

Implements

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [▶ 954]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'. Structs are not supported.

Reference

AdsConnection Class [▶ 357]

TryReadValue Overload [▶ 516]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadValue.T. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: T. The value.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code

Implements

IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.) [▶ 955]

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

AdsConnection Class [▶ 357]

TryReadValue Overload [▶ 516]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadValue(
    string name,
    Type type,
    out Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.
value	Type: System.Object. The read value of the Symbol.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Implements

IAdsSymbolicAccess.TryReadValue(String, Type, Object.) [▶ 956]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

The parameter type must have the same layout as the ADS symbol.

Reference

AdsConnection Class [▶ 357]

TryReadValue Overload [▶ 516]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.50 AdsConnection.TryReadWrite Method

Overload List

	Name	Description
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 521]	
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 522]	

	Name	Description
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte, ReadOnlyMemory) [▶ 522]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadWrite Method (UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryReadWrite Overload [▶ 520]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryReadWrite Overload [▶ 520]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryReadWrite Method (UInt32, UInt32, Memory`1, Void, Byte, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
```

```
    byte timeout,
    ReadOnlyMemory readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
timeout	Type: System.Byte
readBytes	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryReadWrite Overload [▶ 520]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.51 AdsConnection.TryWrite Method

Overload List

	Name	Description
	TryWrite(UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 524]	
	TryWrite(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 525]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWrite Method (UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryWrite Overload [▶ 523]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWrite Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryWrite Overload [▶ 523]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWrite Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryWrite Overload [▶ 523]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.52 AdsConnection.TryWriteControl Method

Overload List

	Name	Description
	TryWriteControl(StateInfo) [▶ 526]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, Int32) [▶ 527]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 528]	
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 528]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsStateControl.TryWriteControl(StateInfo) [▶ 917]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TryWriteControl Overload [▶ 526]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    int timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.
timeout	Type: System.Int32 The timeout.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsStateControlTimeout.TryWriteControl(StateInfo, Int32) [▶ 925]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TryWriteControl Overload [▶ 526]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryWriteControl Overload [▶ 526]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsConnection Class [▶ 357]

TryWriteControl Overload [▶ 526]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.53 AdsConnection.TryWriteValue Method

Overload List

	Name	Description
	TryWriteValue(String, Object) [▶ 529]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(String, T) [▶ 530]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 531]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 532]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue(String, Object) [▶ 957]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TryWriteValue Overload [▶ 529]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue<T>(
    string name,
    T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue.T.(String, T) [▶ 958]

Reference

AdsConnection Class [▶ 357]

TryWriteValue Overload [▶ 529]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [▶ 959]

Reference

AdsConnection Class [▶ 357]

TryWriteValue Overload [▶ 529]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.TryWriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Implements

IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [▶ 960]

Reference

AdsConnection Class [▶ 357]

TryWriteValue Overload [▶ 529]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.54 AdsConnection.UnregisterAdsStateChangedAsync Method

Registers for AdsStateChanged [▶ 567] events as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> UnregisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be unregistered.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateProvider.UnregisterAdsStateChangedAsync(EventHandler.AdsStateChangedEventArgs., CancellationToken) [▶ 933]

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.55 AdsConnection.UnregisterSymbolVersionChangedAsync Method

Unregisters the symbol version changed asynchronous.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to unregister.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsSymbolChangedProvider.UnregisterSymbolVersionChangedAsync(EventHandler.AdsSymbolVersionChangedEventArgs., CancellationToken) [▶ 936]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.56 AdsConnection.Write Method

Overload List

	Name	Description
	Write(UInt32, ReadOnlyMemory) [▶ 535]	
	Write(UInt32, UInt32) [▶ 535]	Trigger Client Method/Command.
	Write(UInt32, UInt32, Int32) [▶ 536]	Trigger Client Method/Command.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 537]	
	Write(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 537]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Write Method (UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

AdsConnection Class [▶ 357]

Write Overload [▶ 534]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Write Method (UInt32, UInt32)

Trigger Client Method/Command.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.

Implements

IAdsReadWrite2.Write(UInt32, UInt32) [▶ 878]

Remarks

This method is used to trigger Client Methods/Commands without parameters.

Reference

AdsConnection Class [▶ 357]

Write Overload [▶ 534]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Write Method (UInt32, UInt32, Int32)

Trigger Client Method/Command.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset,
    int timeout
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
timeout	Type: System.Int32 The timeout.

Remarks

This method is used to trigger Client Methods/Commands without parameters.

Reference

AdsConnection Class [▶ 357]

Write Overload [▶ 534]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

AdsConnection Class [▶ 357]

Write Overload [▶ 534]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.Write Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Reference

AdsConnection Class [▶ 357]

Write Overload [▶ 534]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.57 AdsConnection.WriteAny Method

Overload List

	Name	Description
	WriteAny(UInt32, Object) [▶ 538]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 539]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 540]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 540]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object, .Int32., Int32) [▶ 541]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAny Method (UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint variableHandle,
    Object value
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.

Implements

IAdsAnyAccess.WriteAny(UInt32, Object) [▶ 726]

Reference

AdsConnection Class [▶ 357]

WriteAny Overload [▶ 538]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAny Method (UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Implements

IAdsAnyAccess.WriteAny(UInt32, Object, .Int32.) [▶ 727]

Reference

AdsConnection Class [▶ 357]

WriteAny Overload [▶ 538]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAny Method (UInt32, UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.

Implements

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object) [▶ 728]

Reference

AdsConnection Class [▶ 357]

WriteAny Overload [▶ 538]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAny Method (UInt32, UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Implements

IAdsAnyAccess.WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]

Reference

AdsConnection Class [▶ 357]

WriteAny Overload [▶ 538]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAny Method (UInt32, UInt32, Object, .Int32., Int32)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    int timeout
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
timeout	Type: System.Int32 The timeout.

Implements

IAdsReadWriteTimeoutAccess.WriteAny(UInt32, UInt32, Object, .Int32., Int32) [▶ 885]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

WriteAny Overload [▶ 538]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.58 AdsConnection.WriteAnyAsync Method

Overload List

	Name	Description
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 542]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 543]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 545]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 545]	write any as an asynchronous operation.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyAsync Method (UInt32, Object, CancellationToken)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

WriteAnyAsync Overload [▶ 542]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyAsync Method (UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Remarks

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

AdsConnection Class [▶ 357]

WriteAnyAsync Overload [▶ 542]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]

Reference

AdsConnection Class [▶ 357]

WriteAnyAsync Overload [▶ 542]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)

write any as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Implements

IAdsAnyAccess.WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

WriteAnyAsync Overload [▶ 542]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.59 AdsConnection.WriteAnyStringAsync Method

Overload List

	Name	Description
	WriteAnyStringAsync(String, String, Int32, Encoding, CancellationToken) [▶ 547]	write any string as an asynchronous operation.
	WriteAnyStringAsync(UInt32, String, Int32, Encoding, CancellationToken) [▶ 548]	write any string as an asynchronous operation.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyStringAsync Method (String, String, Int32, Encoding, CancellationToken)

write any string as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyStringAsync(
    string symbolPath,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol path.
value	Type: System.String The value.
length	Type: System.Int32 The length of the string to write
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsConnection Class [▶ 357]

WriteAnyStringAsync Overload [▶ 547]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAnyStringAsync Method (UInt32, String, Int32, Encoding, CancellationToken)

write any string as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAnyStringAsync(
    uint variableHandle,
    string value,
    int length,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
value	Type: System.String The value.
length	Type: System.Int32 The length of the string to write
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task<ResultWrite>.

Remarks

ATTENTION: Potentially this method is unsafe because following data can be overwritten after the string symbol. Please be sure to specify the string length lower than the string size reserved within the process image! The String is written with the specified encoding.

Reference

AdsConnection Class [▶ 357]

WriteAnyStringAsync Overload [▶ 547]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.60 AdsConnection.WriteAsync Method

Overload List

	Name	Description
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 549]	
	WriteAsync(UInt32, UInt32, CancellationToken) [▶ 550]	Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 551]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAsync Method (UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

variableHandle Type: System.UInt32

writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

AdsConnection Class [▶ 357]

WriteAsync Overload [▶ 549]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAsync Method (UInt32, UInt32, CancellationToken)

Triggers a write call at the specified IndexGroup/IndexOffset asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'ReadWrite' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

AdsConnection Class [▶ 357]

WriteAsync Overload [▶ 549]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteAsync Method (UInt32, UInt32, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

AdsConnection Class [▶ 357]

WriteAsync Overload [▶ 549]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.61 AdsConnection.WriteControl Method

Overload List

	Name	Description
	WriteControl(StateInfo) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, Int32) [▶ 552]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 553]	

	Name	Description
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 554]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Implements

IAdsStateControl.WriteControl(StateInfo) [▶ 918]

Reference

AdsConnection Class [▶ 357]

WriteControl Overload [▶ 551]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo,
    int timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.
timeout	Type: System.Int32 The timeout.

Implements

IAdsStateControlTimeout.WriteControl(StateInfo, Int32) [▶ 926]

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

WriteControl Overload [▶ 551]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Reference

AdsConnection Class [▶ 357]

WriteControl Overload [▶ 551]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Reference

AdsConnection Class [▶ 357]

WriteControl Overload [▶ 551]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.62 AdsConnection.WriteControlAsync Method

Overload List

	Name	Description
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 555]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 555]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

state	Type: TwinCAT.Ads.AdsState [▶ 626] The ADS state.
deviceState	Type: System.UInt16 The device state.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'WriteControl' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Implements

IAdsStateControl.WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]

Reference

AdsConnection Class [▶ 357]

WriteControlAsync Overload [▶ 554]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAds> WriteControlAsync(
    AdsState adsState,
    ushort deviceState,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultAds [▶ 989].

Reference

AdsConnection Class [▶ 357]

WriteControlAsync Overload [▶ 554]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.63 AdsConnection.WriteSymbolAsync Method

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteSymbolAsync(
    string name,
    Object value,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.64 AdsConnection.WriteValue Method

Overload List

	Name	Description
	WriteValue(String, Object) [▶ 557]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue.T.(String, T) [▶ 558]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 559]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(ISymbol, T) [▶ 560]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Implements

IAdsSymbolicAccess.WriteValue(String, Object) [▶ 961]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

WriteValue Overload [▶ 557]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue<T>(
    string name,
    T value
)
```

Parameters

name	Type: System.String Name of the ADS symbol.

value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	the value type.

Implements

IAdsSymbolicAccess.WriteValue.T.(String, T) [▶ 961]

Reference

AdsConnection Class [▶ 357]

WriteValue Overload [▶ 557]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Implements

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [▶ 962]

Reference

AdsConnection Class [▶ 357]

WriteValue Overload [▶ 557]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Implements

IAdsSymbolicAccess.WriteValue.T.(ISymbol, T) [▶ 963]

Reference

AdsConnection Class [▶ 357]

WriteValue Overload [▶ 557]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.2.65 AdsConnection.WriteValueAsync Method

Overload List

	Name	Description
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 561]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 562]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

	Name	Description
	WriteValueAsync.T. (ISymbol, T, CancellationToken) [▶ 563]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValueAsync.T. Method (String, T, CancellationToken)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync<T>(
    string name,
    T value,
    CancellationToken cancel
)
```

Parameters

name	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]

Reference

AdsConnection Class [▶ 357]

WriteValueAsync Overload [▶ 560]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValueAsync Method (ISymbol, Object, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. Array and structures are not supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultWrite [▶ 1032].
Task.

Implements

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]

Exceptions

Exception	Condition
ObjectDisposedException	
ClientNotConnectedException [▶ 63]	

Reference

AdsConnection Class [▶ 357]

WriteValueAsync Overload [▶ 560]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.WriteValueAsync.T. Method (ISymbol, T, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Implements

IAdsSymbolicAccess.WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]

Reference

AdsConnection Class [▶ 357]

WriteValueAsync Overload [▶ 560]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3 AdsConnection Events

The AdsConnection [▶ 357] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 564]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 565]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 566]	Occurs when the ADS devices sends a notification to the client.
	AdsStateChanged [▶ 567]	Occurs when ADS State has been changed.
	AdsSumNotification [▶ 568]	Occurs when Notifications are send (bundled notifications)
	AdsSymbolVersionChanged [▶ 569]	Occurs when the symbol version has been changed.
	ConnectionStateChanged [▶ 570]	Occurs when connection status of the AdsConnection [▶ 357] has been changed.
	RouterStateChanged [▶ 571]	Occurs when [router state changed].

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3.1 AdsConnection.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationEventArgs> AdsNotification
```

Value

Type: System.EventHandler.AdsNotificationEventArgs [▶ 597].

Implements

IAdsNotifications.AdsNotification [▶ 866]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

Examples

Example of receiving AdsNotification [▶ 866] events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification2;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];
    int size = sizeof(UInt32);

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new Notific
ationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3.2 AdsConnection.AdsNotificationError Event

Occurs when a exception has occurred during notification management.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError
```

Value

Type: System.EventHandler.AdsNotificationErrorEventArgs [▶ 595].

Implements

IAdsNotifications.AdsNotificationError [▶ 867]

Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the AdsInvalidNotificationException [▶ 591] which describes the invalid notification handle by its Handle [▶ 593] property.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.AdsInvalidNotificationException [▶ 591]

AdsConnection.AdsNotification [▶ 564]

AdsConnection.AdsNotificationEx [▶ 566]

6.2.5.3.3 AdsConnection.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

Value

Type: System.EventHandler.AdsNotificationExEventArgs [▶ 601].

Implements

IAdsNotifications.AdsNotificationEx [▶ 868]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshallers.

Examples

Example of receiving AdsNotificationEx [▶ 868] events.

Trigger on changed values by ADS Notifications

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new Notifica
tionSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint),null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle,ca
ncel); // Unregister Event
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3.4 AdsConnection.AdsStateChanged Event

Occurs when ADS State has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

Value

Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627].

Implements

IAdsStateProvider.AdsStateChanged [▶ 934]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

This event occurs asynchronously if the synchronized flag is not set.

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3.5 AdsConnection.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```

Value

Type: System.EventHandler.AdsSumNotificationEventArgs.

Implements

IAdsNotifications.AdsSumNotification [▶ 868]

Exceptions

Exception	Condition
ObjectDisposedException	

Remarks

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The AdsNotification [▶ 866] and AdsNotificationEx [▶ 868] events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification [▶ 868] event. The full performance is reached only, when all notifications are handled on this event.

Examples

Example of receiving AdsSumNotification [▶ 868] events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsSumNotification += Client_SumNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), n
ew NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;

    foreach(Notification notification in notifications)
    {
    // Notifications can be handled more efficiently, because they occur togeterh
    // handler and can be transformed/
synchronized in one step compared to AdsClient.AdsNotifcation events.
    }
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

6.2.5.3.6 AdsConnection.AdsSymbolVersionChanged Event

Occurs when the symbol version has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

Value

Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637].

Implements

IAdsSymbolChangedProvider.AdsSymbolVersionChanged [▶ 937]

Remarks

This is the case when the connected ADS server restarts. This invalidates all actual opened symbol handles. The SymbolVersion counter doesn't trigger, when an online change is made on the PLC (ports 801, ..., 851 ...)

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.5.3.7 AdsConnection.ConnectionStateChanged Event

Occurs when connection status of the AdsConnection [▶ 357] has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ConnectionStateChangedEventArgs> ConnectionStateChanged
```

Value

Type: System.EventHandler.ConnectionStateChangedEventArgs [▶ 68].

Implements

IConnectionStateProvider.ConnectionStateChanged [▶ 86]

Remarks

The Connection state changes only if the IConnection [▶ 74] is established / shut down or active communication is triggered by the User of the IConnection [▶ 74] object.

Examples

The following sample shows how to keep the ConnectionState [▶ 379] updated by triggering ADS Communication.

Trigger ConnectionState changes in WPF Applications

```
private DispatcherTimer _timer = null;
private AdsSession _session = null;
//private AdsConnection _connection = null;

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _session = new AdsSession(AmsNetId.Local, 10000);
    IConnection connection = _session.Connect();
    tbConnectionState.Text = connection.ConnectionState.ToString();
    _session.ConnectionStateChanged += _session_ConnectionStateChanged;

    _timer = new DispatcherTimer();
    _timer.Interval = TimeSpan.FromMilliseconds(200);
    _timer.Tick += TimerOnTick;

    _timer.Start();
}

private void Window_Unloaded(object sender, RoutedEventArgs e)
{
    _timer.Stop();
    _session.Dispose();
}

private void _session_ConnectionStateChanged(object sender, TwinCAT.ConnectionStateChangedEventArgs
e)
{
    // ConnectionStateChanged will be triggered by communication Invokes
    tbConnectionState.Text = e.NewState.ToString();
}

private void TimerOnTick(object sender, EventArgs eventArgs)
{
    // The Timer Event will occur here in the UI thread because its an DispatcherTimer event!
    // An active ADS request will trigger Connection State periodically!
    StateInfo stateInfo;
    if (_session.Connection.TryReadState(out stateInfo) == AdsErrorCode.NoError)
    {
    tbAdsState.Text = stateInfo.AdsState.ToString();
    }
    else
    {
    tbAdsState.Text = "Invalid";
    }
}
```

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

AdsConnection.ConnectionState [▶ 379]

6.2.5.3.8 AdsConnection.RouterStateChanged Event

Occurs when [router state changed].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```

Value

Type: System.EventHandler.AmsRouterNotificationEventArgs [▶ 695].

Exceptions

Exception	Condition
ObjectDisposedException	

Reference

AdsConnection Class [▶ 357]

TwinCAT.Ads Namespace [▶ 151]

6.2.6 AdsDataTypeArrayInfo Class

Array definition for a single dimension.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsDataTypeArrayInfo
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsDataTypeArrayInfo
```

The AdsDataTypeArrayInfo type exposes the following members.

Properties

	Name	Description
	Elements [▶ 573]	Gets the number of elements.
	LowerBound [▶ 573]	Gets the lower bound.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.6.1 AdsDataTypeArrayInfo Properties

The AdsDataTypeArrayInfo [▶ 572] type exposes the following members.

Properties

	Name	Description
	Elements [▶ 573]	Gets the number of elements.
	LowerBound [▶ 573]	Gets the lower bound.

Reference

AdsDataTypeArrayInfo Class [▶ 572]

TwinCAT.Ads Namespace [▶ 151]

6.2.6.1.1 AdsDataTypeArrayInfo.Elements Property

Gets the number of elements.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Elements { get; }
```

Property Value

Type: Int32

Reference

AdsDataTypeArrayInfo Class [▶ 572]

TwinCAT.Ads Namespace [▶ 151]

6.2.6.1.2 AdsDataTypeArrayInfo.LowerBound Property

Gets the lower bound.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int LowerBound { get; }
```

Property Value

Type: Int32

Reference

AdsDataTypeArrayInfo Class [▶ 572]

TwinCAT.Ads Namespace [▶ 151]

6.2.6.2 AdsDataTypeArrayInfo Methods

The AdsDataTypeArrayInfo [▶ 572] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsDataTypeArrayInfo Class [▶ 572]

TwinCAT.Ads Namespace [▶ 151]

6.2.7 AdsDataTypeId Enumeration

ADS data types.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AdsDataTypeId
```

Members

	Member name	Value	Description
	ADST_VOID	0	Empty Type (0)
	ADST_INT8	16	Integer 8 Bit (16)
	ADST_UINT8	17	Unsigned integer 8 Bit (17)
	ADST_INT16	2	Integer 16 Bit (2)
	ADST_UINT16	18	Unsigned integer 16 Bit (18)
	ADST_INT32	3	Integer 32 Bit (3)
	ADST_UINT32	19	Unsigned Integer 32 Bit (19)
	ADST_INT64	20	LONG Integer 64 Bit (20)
	ADST_UINT64	21	Unsigned Long integer 64 Bit (21)
	ADST_REAL32	4	Real (32 Bit) (4)
	ADST_REAL64	5	Real 64 Bit (5)
	ADST_BIGTYPE	65	Blob (65)
	ADST_STRING	30	STRING (30)
	ADST_WSTRING	31	WSTRING (31)
	ADST_REAL80	32	ADS REAL80 (32)
	ADST_BIT	33	ADS BIT (33)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.8 AdsErrorCode Enumeration

Describes the ADS error that occurred.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AdsErrorCode
```

Members

	Member name	Value	Description
	None	-1	None / Uninitialized
	NoError	0	No Error / Succeeded. Error code: 0(0x000).
	Succeeded	0	No Error / Succeeded. Error code: 0(0x000).
	InternalError	1	Internal Error. Error code: 1(0x001).
	NoRTime	2	No Realtime. Error code: 2(0x002).
	LockedMemoryError	3	Allocation locked memory error. Error code: 3(0x003).

	Member name	Value	Description
	MailBoxError	4	Insert mailbox error. Error code: 4(0x004).
	WrongHMsg	5	Wrong receive HMSG. Error code: 5(0x005).
	TargetPortNotFound	6	Target port not found. Error code: 6(0x006).
	TargetMachineNotFound	7	Target machine not found. Error code: 7(0x007).
	UnknownCommandID	8	Unknown command ID. Error code: 8(0x008).
	BadTaskID	9	Bad task ID. Error code: 9(0x009).
	NoIO	10	No IO. Error code: 10(0x00A).
	UnknownAmsCommand	11	Unknown AMS command. Error code: 11(0x00B).
	Win32Error	12	Win 32 error. Error code: 12(0x00C).
	PortNotConnected	13	Port is not connected. Error code: 13(0x00D).
	InvalidAmsLength	14	Invalid AMS length. Error code: 14(0x00E).
	InvalidAmsNetID	15	Invalid AMS Net ID. Error code: 15(0x00F).
	LowInstallLevel	16	Low Installation level. Error code: 16(0x010).
	NoDebug	17	No debug available. Error code: 17(0x011).
	PortDisabled	18	Port disabled. Error code: 18(0x012).
	PortConnected	19	Port is already connected. Error code: 19(0x013).
	AmsSyncWin32Error	20	AMS Sync Win32 error. Error code: 20(0x014).
	SyncTimeOut	21	AMS Sync timeout. Error code: 21(0x015).
	AmsSyncAmsError	22	AMS Sync AMS error Error code: 22(0x016).
	AmsSyncNoIndexMap	23	AMS Sync no index map. Error code: 23(0x017).
	InvalidAmsPort	24	Invalid AMS port. Error code: 24(0x018).
	NoMemory	25	No memory. Error code: 25(0x019).
	TCPSendError	26	TCP send error. Error code: 26(0x01A).

	Member name	Value	Description
	HostUnreachable	27	Host unreachable. Error code: 27(0x1B).
	AmsInvalidFragment	28	Invalid AMS fragment. Error code: 28(0x1C).
	NoLockedMemory	1280	Router: no locked memory. Error code: 1280(0x500).
	ResizeMemory	1281	Router: The size of the router memory could not be changed. Error code: 1281(0x501).
	MailboxFull	1282	Router: mailbox full. Error code: 1282(0x502).
	DebugBoxFull	1283	Router: The mailbox has reached the maximum number of possible messages. Error code: 1283(0x503).
	UnknownPortType	1284	Router: Unknown Port Type Error code: 1284(0x504).
	RouterNotInitialized	1285	Router: Router is not initialized. Error code: 1285(0x505).
	PortAlreadyInUse	1286	Router: The desired port number is already assigned. Error code: 1286(0x506).
	PortNotRegistered	1287	Router: Port not registered. Error code: 1287(0x507).
	NoMoreQueues	1288	Router: The maximum number of Ports reached. Error code: 1288(0x508).
	InvalidPort	1289	Router: The port is invalid. Error code: 1289(0x509).
	RouterNotActive	1290	Router: TwinCAT Router not active. Error code: 1290(0x50A).
	DeviceError	1792	error class <device error"> Error code: 1792(0x700).
	DeviceServiceNotSupported	1793	Service is not supported by server. Error code: 1793(0x701).
	DeviceInvalidGroup	1794	Invalid index group. Error code: 1794(0x702).
	DeviceInvalidOffset	1795	Invalid index offset. Error code: 1795(0x703).
	DeviceInvalidAccess	1796	Reading/writing not permitted. Error code: 1796(0x704).
	DeviceInvalidSize	1797	Parameter size not correct. Error code: 1797(0x705).
	DeviceInvalidData	1798	Invalid parameter value(s). Error code: 1798(0x706).
	DeviceNotReady	1799	Device is not in a ready state. Error code: 1799(0x707).

	Member name	Value	Description
	DeviceBusy	1800	Device is busy. Error code: 1800(0x708).
	DeviceInvalidContext	1801	Invalid context (must be in Windows). Error code: 1801(0x709).
	DeviceNoMemory	1802	Out of memory. Error code: 1802(0x70a).
	DeviceInvalidParam	1803	Invalid parameter value(s). Error code: 1803(0x70b).
	DeviceNotFound	1804	Not found(files, ...). Error code: 1804(0x70c).
	DeviceSyntaxError	1805	Syntax error in command or file. Error code: 1805(0x70d).
	DeviceIncompatible	1806	Objects do not match. Error code: 1806(0x70e).
	DeviceExists	1807	Object already exists. Error code: 1807(0x70f).
	DeviceSymbolNotFound	1808	Symbol not found. Error code: 1808(0x7010).
	DeviceSymbolVersionInvalid	1809	Symbol version is invalid. Error code: 1809(0x711).
	DeviceInvalidState	1810	Server is not i a valid state. Error code: 1810(0x712).
	DeviceTransModeNotSupported	1811	ADS transmode is not supported. Error code: 1811(0x713).
	DeviceNotifyHandleInvalid	1812	Notification handle is invalid. Error code: 1812(0x714).
	DeviceClientUnknown	1813	Notification vlient not registered. Error code: 1813(0x715).
	DeviceNoMoreHandles	1814	No more notification handles. Error code: 1814(0x716).
	DeviceInvalidWatchsize	1815	Size for watch to big. Error code: 1815(0x717).
	DeviceNotInitialized	1816	Device is not initialized. Errocr code: 1818(0x718).
	DeviceTimeOut	1817	Devicee has a timeout. Error code: 1817(0x719).
	DeviceNoInterface	1818	Query interface has failed. Error code: 1818(0x71A).
	DeviceInvalidInterface	1819	Wrong interface required. Error code: 1819(0x71B).
	DeviceInvalidCLSID	1820	Class ID is invalid. Error code: 1820(0x71C).
	DeviceInvalidObjectID	1821	Object ID is invalid. Error code: 1821(0x71D).
	DeviceRequestIsPending	1822	Device: Request is Pending. Error code: 1822(0x71E).

	Member name	Value	Description
	DeviceRequestIsAborted	1823	Device: Request is Aborted. Error code: 1823(0x71F).
	DeviceSignalWarning	1824	Device: Signal warning. Error code: 1824(0x720).
	DeviceInvalidArrayIndex	1825	Device: Invalid Array Index (ADSERR_DEVICE_INVALIDARRAYIDX) Error code: 1825(0x721).
	DeviceSymbolNotActive	1826	Device: Symbol not Active Error code: 1826(0x722).
	DeviceAccessDenied	1827	Device: Access denied. Error code: 1827(0x723).
	DeviceLicenseNotFound	1828	Device: license not found Error code: 1828(0x724).
	DeviceLicenseExpired	1829	Device: license expired Error code: 1829(0x725).
	DeviceLicenseExceeded	1830	Device: license exceeded Error code: 1830(0x726).
	DeviceLicenseInvalid	1831	Device: license invalid Error code: 1831(0x727).
	DeviceLicenseSystemId	1832	Device: license invalid system id Error code: 1832(0x728).
	DeviceLicenseNoTimeLimit	1833	Device: license not time limited Error code: 1833(0x729).
	DeviceLicenseFutureIssue	1834	Device: license issue time in the future Error code: 1834(0x72A).
	DeviceLicenseTimeToLong	1835	Device: license time period to long Error code: 1835(0x72B).
	DeviceException	1836	Device: Exception in device specific code Error code: 1836(0x72C).
	DeviceLicenseDuplicated	1837	Device: license file read twice Error code: 1837(0x72D).
	DeviceSignatureInvalid	1838	Device: invalid signature Error code: 1838(0x72E).
	DeviceCertificateInvalid	1839	Device: public key certificate Error code: 1839(0x72F).
	DeviceLicenseOemNotFound	1840	Device: public key of OEM unknown Error code: 1840(0x730).
	DeviceLicenseRestricted	1841	Device: license not valid for this system id type Error code: 1841(0x731).
	DeviceLicenseDemoDenied	1842	Device: trial license denied Error code: 1842(0x732).
	DeviceInvalidFncId	1843	Device: function id is invalid Error code: 1843(0x733).

	Member name	Value	Description
	DeviceOutOfRange	1844	Device: a parameter, an index, an iterator, ... is out of range Error code: 1844(0x734).
	DeviceInvalidAlignment	1845	Device: invalid alignment Error code: 1845(0x735).
	DeviceLicensePlatform	1846	Device: license invalid platform level Error code: 1846(0x736).
	ClientError	1856	Error class <client error> Error code: 1856(0x740).
	ClientInvalidParameter	1857	Parameter at service is invalid. Error code: 1857(0x741).
	ClientListEmpty	1858	Polling list is empty. Error code: 1858(0x742).
	ClientVariableInUse	1859	Variable connection is already in use. Error code: 1859(0x743).
	ClientDuplicateInvokeID	1860	Invoke ID already in use. Error code: 1860(0x744).
	ClientSyncTimeOut	1861	Timeout has elapsed. Error code: 1861(x745).
	ClientW32OR	1862	Error in win32 subsystem. Error code: 1862(0x746).
	ClientTimeoutInvalid	1863	Timeout value is invalid. Error code: 1863(0x747).
	ClientPortNotOpen	1864	ADS port is not opened. Error code: 1864(0x748).
	ClientNoAmsAddr	1865	No AMS Address. Error code: 1865(0x749).
	ClientSyncInternal	1872	An internal in ADS sync has occurred. Error code: 1872(0x750).
	ClientAddHash	1873	Hash table overflow. Error code: 1873(0x751).
	ClientRemoveHash	1874	There are no more symbols in the hash table. Error code: 1874(0x752).
	ClientNoMoreSymbols	1875	There are no more symbols in cache. Error code: 1875(0x753).
	ClientSyncResInvalid	1876	An invalid response has been received. Error code: 1876(0x754).
	ClientSyncPortLocked	1877	Sync port is locked. Error code: 1877(0x755).
	ClientQueueFull	32768	Client queue is full
	WSA_ConnRefused	10061	Windows sockets connection refused (0x274d, 10061)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.9 AdsErrorCodeExtensions Class

Class AdsErrorCodeExtensions.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsErrorCodeExtensions
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class AdsErrorCodeExtensions
```

The AdsErrorCodeExtensions type exposes the following members.

Methods

	Name	Description
	Failed [▶ 582]	Indicates, that the communication failed with AdsErrorCode [▶ 575].
	Succeeded [▶ 582]	Indicates, that the communication / AdsErrorCode [▶ 575] doesn't show an error.

Remarks

This class extends the AdsErrorCode [▶ 575] by Succeeded(AdsErrorCode) [▶ 582] and Failed(AdsErrorCode) [▶ 582] methods.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.9.1 AdsErrorCodeExtensions Methods

The AdsErrorCodeExtensions [▶ 581] type exposes the following members.

Methods

	Name	Description
	Failed [▶ 582]	Indicates, that the communication failed with AdsErrorCode [▶ 575].
	Succeeded [▶ 582]	Indicates, that the communication / AdsErrorCode [▶ 575] doesn't show an error.

Reference

AdsErrorCodeExtensions Class [▶ 581]

TwinCAT.Ads Namespace [▶ 151]

6.2.9.1.1 AdsErrorCodeExtensions.Failed Method

Indicates, that the communication failed with AdsErrorCode [▶ 575].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool Failed(
    this AdsErrorCode errorCode
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type AdsErrorCode [▶ 575]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsErrorCodeExtensions Class [▶ 581]

TwinCAT.Ads Namespace [▶ 151]

6.2.9.1.2 AdsErrorCodeExtensions.Succeeded Method

Indicates, that the communication / AdsErrorCode [▶ 575] doesn't show an error.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool Succeeded(
    this AdsErrorCode errorCode
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.

Return Value

Type: Boolean
true if the AdsErrorCode [▶ 575] is NoError [▶ 575], false otherwise.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type AdsErrorCode [▶ 575]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsErrorCodeExtensions Class [▶ 581]

TwinCAT.Ads Namespace [▶ 151]

6.2.10 AdsErrorException Class

The exception that is thrown when an ADS error occurs.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.Ads.AdsErrorException
        TwinCAT.Ads.AdsSumCommandException [▶ 632]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class AdsErrorException : AdsException
```

The AdsErrorException type exposes the following members.

Constructors

	Name	Description
	AdsErrorException. [▶ 585]	Initializes a new Instance of the AdsErrorException class.
	AdsErrorException(SerializationInfo, StreamingContext) [▶ 586]	Initializes a new instance of the AdsErrorException class.

	Name	Description
	AdsErrorException(String, AdsErrorCode) [▶ 586]	Initializes a new Instance of the AdsErrorException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 587]	Gets the error code of the Exception.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
S	Create(AdsErrorCode) [▶ 589]	Creates the AdsErrorException
S	Create(String, AdsErrorCode) [▶ 589]	Creates the AdsErrorException
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 590]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.10.1 AdsErrorException Constructor

Overload List

	Name	Description
	AdsErrorException. [▶ 585]	Initializes a new Instance of the AdsErrorException class.
	AdsErrorException(SerializationInfo, StreamingContext) [▶ 586]	Initializes a new instance of the AdsErrorException [▶ 583] class.
	AdsErrorException(String, AdsErrorCode) [▶ 586]	Initializes a new Instance of the AdsErrorException class.

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.1.1 AdsErrorException Constructor

Initializes a new Instance of the AdsErrorException class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorException()
```

Reference

AdsErrorException Class [▶ 583]

AdsErrorException Overload [▶ 585]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.1.2 AdsErrorException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the AdsErrorException [▶ 583] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorException(
    SerializationInfo info,
    StreamingContext streamingContext
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

AdsErrorException Class [▶ 583]

AdsErrorException Overload [▶ 585]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.1.3 AdsErrorException Constructor (String, AdsErrorCode)

Initializes a new Instance of the AdsErrorException class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorException(
    string message,
    AdsErrorCode errorCode
)
```

Parameters

message	Type: System.String The message.
errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.

Reference

AdsErrorException Class [▶ 583]

AdsErrorException Overload [▶ 585]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.2 AdsErrorException Properties

The AdsErrorException [▶ 583] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 587]	Gets the error code of the Exception.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.2.1 AdsErrorException.ErrorCode Property

Gets the error code of the Exception.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode ErrorCode { get; }
```

Property Value

Type: AdsErrorCode [▶ 575]
The error code.

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.3 AdsErrorException Methods

The AdsErrorException [▶ 583] type exposes the following members.

Methods

	Name	Description
	Create(AdsErrorCode) [▶ 589]	Creates the AdsErrorException
	Create(String, AdsErrorCode) [▶ 589]	Creates the AdsErrorException
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 590]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.3.1 AdsErrorException.Create Method

Overload List

	Name	Description
	Create(AdsErrorCode) [▶ 589]	Creates the AdsErrorException
	Create(String, AdsErrorCode) [▶ 589]	Creates the AdsErrorException

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

AdsErrorException.Create Method (AdsErrorCode)

Creates the AdsErrorException

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsErrorException Create(
    AdsErrorCode adsErrorCode
)
```

Parameters

adsErrorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ads error code.

Return Value

Type: AdsErrorException [▶ 583]
AdsErrorException.

Reference

AdsErrorException Class [▶ 583]

Create Overload [▶ 589]

TwinCAT.Ads Namespace [▶ 151]

AdsErrorException.Create Method (String, AdsErrorCode)

Creates the AdsErrorException

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsErrorException Create(
    string message,
    AdsErrorCode adsErrorCode
)
```

Parameters

message	Type: System.String The message.
adsErrorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ads error code.

Return Value

Type: AdsErrorException [▶ 583]
AdsErrorException.

Reference

AdsErrorException Class [▶ 583]

Create Overload [▶ 589]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.3.2 AdsErrorException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.10.4 AdsErrorException Events

The AdsErrorException [▶ 583] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

AdsErrorException Class [▶ 583]

TwinCAT.Ads Namespace [▶ 151]

6.2.11 AdsInvalidNotificationException Class

This AdsInvalidNotificationException is created if the length of the notification data is 0. This indicates that the notification handle is not valid any more. This exception is passed to the AdsNotificationErrorEvent.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.Ads.AdsInvalidNotificationException

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class AdsInvalidNotificationException : AdsException
```

The AdsInvalidNotificationException type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	Handle [▶ 593]	Handle of the notification.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)
	TimeStamp [▶ 593]	Gets the Time stamp as long

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 594]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.11.1 AdsInvalidNotificationException Properties

The AdsInvalidNotificationException [▶ 591] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)

	Name	Description
	Handle [▶ 593]	Handle of the notification.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)
	TimeStamp [▶ 593]	Gets the Time stamp as long

Reference

AdsInvalidNotificationException Class [▶ 591]

TwinCAT.Ads Namespace [▶ 151]

6.2.11.1.1 AdsInvalidNotificationException.Handle Property

Handle of the notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Handle { get; }
```

Property Value

Type: UInt32
The handle.

Reference

AdsInvalidNotificationException Class [▶ 591]

TwinCAT.Ads Namespace [▶ 151]

6.2.11.1.2 AdsInvalidNotificationException.TimeStamp Property

Gets the Time stamp as long

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The time stamp.

Reference

AdsInvalidNotificationException Class [▶ 591]

TwinCAT.Ads Namespace [▶ 151]

6.2.11.2 AdsInvalidNotificationException Methods

The AdsInvalidNotificationException [▶ 591] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 594]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AdsInvalidNotificationException Class [▶ 591]

TwinCAT.Ads Namespace [▶ 151]

6.2.11.2.1 AdsInvalidNotificationException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

AdsInvalidNotificationException Class [▶ 591]

TwinCAT.Ads Namespace [▶ 151]

6.2.12 AdsNotificationErrorEventArgs Class

Arguments for the AdsNotificationError [▶ 867] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationErrorEventArgs

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsNotificationErrorEventArgs : EventArgs
```

The AdsNotificationErrorEventArgs type exposes the following members.

Properties

	Name	Description
	Exception [▶ 596]	Exception that was caught while handling notifications.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.12.1 AdsNotificationErrorEventArgs Properties

The AdsNotificationErrorEventArgs [▶ 595] type exposes the following members.

Properties

	Name	Description
	Exception [▶ 596]	Exception that was caught while handling notifications.

Reference

AdsNotificationErrorEventArgs Class [▶ 595]

TwinCAT.Ads Namespace [▶ 151]

6.2.12.1.1 AdsNotificationErrorEventArgs.Exception Property

Exception that was caught while handling notifications.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Exception Exception { get; }
```

Property Value

Type: Exception

Reference

AdsNotificationErrorEventArgs Class [▶ 595]

TwinCAT.Ads Namespace [▶ 151]

6.2.12.2 AdsNotificationErrorEventArgs Methods

The AdsNotificationErrorEventArgs [▶ 595] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsNotificationErrorEventArgs Class [▶ 595]

TwinCAT.Ads Namespace [▶ 151]

6.2.13 AdsNotificationEventArgs Class

Event argument class for AdsNotification [▶ 866] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationEventArgs
      TwinCAT.Ads.AdsNotificationExEventArgs [▶ 601]
      TwinCAT.Ads.ValueNotificationEventArgs.T. [▶ 1053]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsNotificationEventArgs : EventArgs,
    INotification
```

The AdsNotificationEventArgs type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value.
	Handle [▶ 599]	Gets the Notification handle.

	Name	Description
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset.
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.13.1 AdsNotificationEventArgs Properties

The AdsNotificationEventArgs [▶ 597] type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value.
	Handle [▶ 599]	Gets the Notification handle.
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset.
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.13.1.1 AdsNotificationEventArgs.Data Property

Memory object holding the Notification Data/Value.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemory Data { get; }
```

Property Value

Type: ReadOnlyMemory

Implements

INotification.Data [▶ 970]

Remarks

This Memory object can be seen as binary 'View' to the value object. It represents exactly the data that corresponds to the Handle [▶ 599].

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.13.1.2 AdsNotificationEventArgs.Handle Property

Gets the Notification handle.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Handle { get; }
```

Property Value

Type: UInt32

Implements

INotification.Handle [▶ 970]

Remarks

The Notification Handle is the handle that is created during ADS Notification registration (AddDeviceNotificationAsync Overload [▶ 847], AddDeviceNotification Overload [▶ 844]) and used for deregistration (DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858],DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]).

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.13.1.3 AdsNotificationEventArgs.TimeStamp Property

Gets the time stamp of this Notification as DateTimeOffset.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset

Implements

INotification.TimeStamp [▶ 971]

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.13.1.4 AdsNotificationEventArgs.UserData Property

Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object UserData { get; }
```

Property Value

Type: Object

Implements

INotification.UserData [▶ 971]

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.13.2 AdsNotificationEventArgs Methods

The AdsNotificationEventArgs [▶ 597] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsNotificationEventArgs Class [▶ 597]

TwinCAT.Ads Namespace [▶ 151]

6.2.14 AdsNotificationExEventArgs Class

Arguments for AdsNotificationEx [▶ 868] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationEventArgs [▶ 597]
      TwinCAT.Ads.AdsNotificationExEventArgs

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsNotificationExEventArgs : AdsNotificationEventArgs
```

The AdsNotificationExEventArgs type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Handle [▶ 599]	Gets the Notification handle. (Inherited from AdsNotificationEventArgs [▶ 597].)
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [▶ 597].)

	Name	Description
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Value [▶ 602]	Value of the ADS Notification.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.14.1 AdsNotificationExEventArgs Properties

The AdsNotificationExEventArgs [▶ 601] type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Handle [▶ 599]	Gets the Notification handle. (Inherited from AdsNotificationEventArgs [▶ 597].)
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [▶ 597].)
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Value [▶ 602]	Value of the ADS Notification.

Reference

AdsNotificationExEventArgs Class [▶ 601]

TwinCAT.Ads Namespace [▶ 151]

6.2.14.1.1 AdsNotificationExEventArgs.Value Property

Value of the ADS Notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object Value { get; }
```

Property Value

Type: Object

Reference

AdsNotificationExEventArgs Class [▶ 601]

TwinCAT.Ads Namespace [▶ 151]

6.2.14.2 AdsNotificationExEventArgs Methods

The AdsNotificationExEventArgs [▶ 601] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsNotificationExEventArgs Class [▶ 601]

TwinCAT.Ads Namespace [▶ 151]

6.2.15 AdsSession Class

AdsSession class

Inheritance Hierarchy

System.Object
  TwinCAT.Session [▶ 101]
    TwinCAT.Ads.AdsSessionBase [▶ 614]
      TwinCAT.Ads.AdsSession

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsSession : AdsSessionBase
```

The AdsSession type exposes the following members.

Constructors

	Name	Description
	AdsSession(AmsAddress) [▶ 609]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsAddress, SessionSettings) [▶ 610]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsNetId, Int32) [▶ 610]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsAddress, SessionSettings, Object) [▶ 611]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsNetId, Int32, SessionSettings) [▶ 611]	Initializes a new instance of the AdsSessionBase [▶ 614] class.

Properties

	Name	Description
	Address [▶ 618]	Gets the target address of the AdsSessionBase [▶ 614] (Inherited from AdsSessionBase [▶ 614].)
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation. (Inherited from Session [▶ 101].)
	Connection [▶ 618]	Gets the connection. (Inherited from AdsSessionBase [▶ 614].)
	ConnectionState [▶ 104]	Gets the current Connection state of the Session [▶ 101] (Inherited from Session [▶ 101].)
	Disposed [▶ 105]	Gets a value indicating whether this Session [▶ 101] is disposed. (Inherited from Session [▶ 101].)
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established. (Inherited from Session [▶ 101].)
	Id [▶ 106]	Gets the Session Identifier (Inherited from Session [▶ 101].)
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected. (Inherited from Session [▶ 101].)
	Name [▶ 107]	Gets the name of the session (Inherited from Session [▶ 101].)
	NetId [▶ 619]	Gets the NetId of the Session (Inherited from AdsSessionBase [▶ 614].)
	Owner [▶ 619]	Gets the Session owner. (Inherited from AdsSessionBase [▶ 614].)
	Port [▶ 620]	Gets the Ams Port of the Session (Inherited from AdsSessionBase [▶ 614].)

	Name	Description
	Settings [▶ 620]	Gets the settings of the connection. (Inherited from AdsSessionBase [▶ 614].)
	Statistics [▶ 621]	Gets the Communication / Session statistics. (Inherited from AdsSessionBase [▶ 614].)
	SymbolServer [▶ 108]	Gets the symbol server. (Inherited from Session [▶ 101].)

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88] (Inherited from Session [▶ 101].)
	Connect [▶ 110]	Connects the session. (Inherited from Session [▶ 101].)
	Disconnect [▶ 111]	Disconnects the session from the target. (Inherited from Session [▶ 101].)
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [▶ 101].)
	Dispose(Boolean) [▶ 622]	Releases unmanaged and - optionally - managed resources. (Inherited from AdsSessionBase [▶ 614].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 623]	Finalizes an instance of the AdsSessionBase [▶ 614] class. (Inherited from AdsSessionBase [▶ 614].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 623]	Gets the name/string identifier of the session. (Inherited from AdsSessionBase [▶ 614].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 624]	Handler function connecting the Session. (Inherited from AdsSessionBase [▶ 614].)
	OnCreateSymbolServer [▶ 624]	Handler function creating the symbol server object. (Inherited from AdsSessionBase [▶ 614].)
	OnDisconnect [▶ 625]	Called when [disconnect]. (Inherited from AdsSessionBase [▶ 614].)
	OnGetAddress [▶ 625]	Handler function getting the address of the session. (Inherited from AdsSessionBase [▶ 614].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from Session [▶ 101].)

Remarks

On top of the well known AdsClient [▶ 154] class that is used traditionally for ADS communication, the AdsSessionBase [▶ 614] class provides the following additionally abilities out of the box: These are used to provide more stable connections to ADS Servers than the AdsClient [▶ 154] can provide. The main issues are Resurrection / Self-Healing after communication timeouts, faster and less error prone reaction to communication errors (not necessarily waiting for communication timeouts) und enhanced communication diagnosis. These enhanced features are provided by the following additions to the TwinCAT.Ads API:

- AdsConnection [▶ 357] class.
- Enhanced diagnosis in form of communication statistics Statistics [▶ 621]
- (semi-automatic) Resurrectable client communication with AdsConnection [▶ 357] objects.
- Symbol caching SymbolServer [▶ 108]
- Fail fast handler for connection stabilization IFailFastHandler

The AdsConnection [▶ 357] is established by calling the Connect. [▶ 110] method. The returned AdsConnection [▶ 357] can be used as long the AdsSessionBase [▶ 614] exists.

Examples

The following sample shows a simple use of the AdsSessionBase [▶ 614] object. The AdsSession object (and the dynamic SymbolLoader features) are only available from .NET 4 and upwards.

Use Session (async)

```
using System;
using System.Diagnostics;
using System.Threading;

using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SessionAsync
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;
        SessionSettings settings = SessionSettings.Default; // Default settings are Async access wit
h Timeout 5 sec

        // Async access is necessary for Console applications!

        using (AdsSession session = new AdsSession(address, settings))
        {
        AdsConnection connection = (AdsConnection) session.Connect(); // Establish the connection
        connection.ConnectionStateChanged += Connection_ConnectionStateChanged;

        ConnectionState connectionState = connection.ConnectionState; // The actual connection state

        // Read the identification and version number of the device
        var result = await connection.ReadDeviceInfoAsync(cancel);

        result.ThrowOnError(); // Throws exception if failed.

        DeviceInfo deviceInfo = result.DeviceInfo;
        Version version = deviceInfo.Version.ConvertToStandard();
        Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
        Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));
```

```
        /// Read the state of the device
        var resultReadState = await connection.ReadStateAsync(cancel);
        resultReadState.ThrowOnError();

        StateInfo stateInfo = resultReadState.State;
        AdsState adsState = stateInfo.AdsState;
        short deviceState = stateInfo.DeviceState;

        Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
        Console.WriteLine(string.Format("AdsState   : {0}", adsState));

        // Other ADS methods (as formerly used on AdsClient) can be used also on connection object:

        // connection.ReadAsync(...)
        // connection.WriteAsync(...)
        // connection.AddDeviceNotificationEx += ...

        // Session communication Diagnostic:

        int resurrectionTries = connection.ResurrectingTries;
        int succeededResurrections = connection.Resurrections;

        AdsCommunicationStatistics statistics = session.Statistics; // The communication statistics

        // Symbol access:
        // The Session holds and Caches the Symbolic data information
        var resultDataTypes = await session.SymbolServer.GetDataTypesAsync(cancel);
        var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);

        if (resultDataTypes.Succeeded && resultSymbols.Succeeded) // Check for succeed
        {
            IDataTypeCollection<IDataType> types = resultDataTypes.DataTypes;
            ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;

            Symbol projectNameSymbol = (Symbol)symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectNa
me"];
            var resultReadProjectName = await projectNameSymbol.ReadValueAsync(cancel);
            string projectName = (string)resultReadProjectName.Value;

            // Or use dynamic objects
            dynamic appInfo = symbols["TwinCAT_SystemInfoVarList._AppInfo"];
            string projectName2 = appInfo.ProjectName; // Property dynamically created (synchronous
call)!
        }
        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
        }
    }

    private static void Connection_ConnectionStateChanged(object sender, ConnectionStateChangedEvent
Args e)
    {
        Console.WriteLine("Connection State changed (NewState: {0}, OldState: {1}",e.NewState,e.OldS
tate);
    }
```

Use Session (sync)

```
using System;
using System.Diagnostics;
using System.Threading;

using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class Session
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
```

```
        AmsAddress address = ArgParser.Parse(args);
        SessionSettings settings = SessionSettings.Default; // Default settings are Async access wit
h Timeout 5 sec

        // Async access is necessary for Console applications!

        using (AdsSession session = new AdsSession(address, settings))
        {
        AdsConnection connection = (AdsConnection) session.Connect(); // Establish the connection
        connection.ConnectionStateChanged += Connection_ConnectionStateChanged;

        ConnectionState connectionState = connection.ConnectionState; // The actual connection state

        // Read the identification and version number of the device
        DeviceInfo deviceInfo = connection.ReadDeviceInfo();
        Version version = deviceInfo.Version.ConvertToStandard();
        Console.WriteLine(string.Format("DeviceName: {0}", deviceInfo.Name));
        Console.WriteLine(string.Format("DeviceVersion: {0}", version.ToString(3)));

        /// Read the state of the device
        StateInfo stateInfo = connection.ReadState();
        AdsState adsState = stateInfo.AdsState;

        short deviceState = stateInfo.DeviceState;
        Console.WriteLine(string.Format("DeviceState: {0}", deviceState));
        Console.WriteLine(string.Format("AdsState   : {0}", adsState));

        // Other ADS methods (as formerly used on AdsClient) can be used also on connection object:

        // connection.Read(...)
        // connection.Write(...)
        // connection.AddDeviceNotificationEx += ...


        // Session communication Diagnostic:

        int resurrectionTries = connection.ResurrectingTries;
        int succeededResurrections = connection.Resurrections;

        AdsCommunicationStatistics statistics = session.Statistics; // The communication statistics

        // Symbol access:
        // The Session holds and Caches the Symbolic data information
        IDataTypeCollection<IDataType> types = session.SymbolServer.DataTypes;
        ISymbolCollection<ISymbol> symbols = session.SymbolServer.Symbols;

        Symbol projectNameSymbol = (Symbol) symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"
];
        string projectName = (string) projectNameSymbol.ReadValue();

        // Or use dynamic objects
        dynamic appInfo = symbols["TwinCAT_SystemInfoVarList._AppInfo"];
        string projectName2 = appInfo.ProjectName;

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
        }
    }

    private static void Connection_ConnectionStateChanged(object sender, ConnectionStateChangedEvent
Args e)
    {
        Console.WriteLine("Connection State changed (NewState: {0}, OldState: {1}",e.NewState,e.OldS
tate);
    }
```

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Session [▶ 101]

TwinCAT.Ads.IAdsSession [▶ 908]

IInterceptionFactory

6.2.15.1 AdsSession Constructor

Overload List

	Name	Description
	AdsSession(AmsAddress) [▶ 609]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsAddress, SessionSettings) [▶ 610]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsNetId, Int32) [▶ 610]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsAddress, SessionSettings, Object) [▶ 611]	Initializes a new instance of the AdsSessionBase [▶ 614] class.
	AdsSession(AmsNetId, Int32, SessionSettings) [▶ 611]	Initializes a new instance of the AdsSessionBase [▶ 614] class.

Reference

AdsSession Class [▶ 603]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.1.1 AdsSession Constructor (AmsAddress)

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSession(
    AmsAddress address
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.

Reference

AdsSession Class [▶ 603]

AdsSession Overload [▶ 609]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.1.2 AdsSession Constructor (AmsAddress, SessionSettings)

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSession(
    AmsAddress address,
    SessionSettings settings
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.
settings	Type: TwinCAT.Ads.SessionSettings [▶ 1035] The settings.

Reference

AdsSession Class [▶ 603]

AdsSession Overload [▶ 609]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.1.3 AdsSession Constructor (AmsNetId, Int32)

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSession(
    AmsNetId netId,
    int port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net identifier.
port	Type: System.Int32 The port.

Reference

AdsSession Class [▶ 603]

AdsSession Overload [▶ 609]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.1.4 AdsSession Constructor (AmsAddress, SessionSettings, Object)

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSession(
    AmsAddress address,
    SessionSettings settings,
    Object owner
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.
settings	Type: TwinCAT.Ads.SessionSettings [▶ 1035] The settings.
owner	Type: System.Object The session owner

Reference

AdsSession Class [▶ 603]

AdsSession Overload [▶ 609]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.1.5 AdsSession Constructor (AmsNetId, Int32, SessionSettings)

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSession(
    AmsNetId netId,
    int port,
    SessionSettings settings
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net identifier.
port	Type: System.Int32 The port.
settings	Type: TwinCAT.Ads.SessionSettings [▶ 1035] The settings.

Reference

AdsSession Class [▶ 603]

AdsSession Overload [▶ 609]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.2 AdsSession Properties

The AdsSession [▶ 603] type exposes the following members.

Properties

	Name	Description
	Address [▶ 618]	Gets the target address of the AdsSessionBase [▶ 614] (Inherited from AdsSessionBase [▶ 614].)
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation. (Inherited from Session [▶ 101].)
	Connection [▶ 618]	Gets the connection. (Inherited from AdsSessionBase [▶ 614].)
	ConnectionState [▶ 104]	Gets the current Connection state of the Session [▶ 101] (Inherited from Session [▶ 101].)
	Disposed [▶ 105]	Gets a value indicating whether this Session [▶ 101] is disposed. (Inherited from Session [▶ 101].)
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established. (Inherited from Session [▶ 101].)
	Id [▶ 106]	Gets the Session Identifier (Inherited from Session [▶ 101].)
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected. (Inherited from Session [▶ 101].)
	Name [▶ 107]	Gets the name of the session (Inherited from Session [▶ 101].)
	NetId [▶ 619]	Gets the NetId of the Session (Inherited from AdsSessionBase [▶ 614].)

	Name	Description
	Owner [▶ 619]	Gets the Session owner. (Inherited from AdsSessionBase [▶ 614].)
	Port [▶ 620]	Gets the Ams Port of the Session (Inherited from AdsSessionBase [▶ 614].)
	Settings [▶ 620]	Gets the settings of the connection. (Inherited from AdsSessionBase [▶ 614].)
	Statistics [▶ 621]	Gets the Communication / Session statistics. (Inherited from AdsSessionBase [▶ 614].)
	SymbolServer [▶ 108]	Gets the symbol server. (Inherited from Session [▶ 101].)

Reference

AdsSession Class [▶ 603]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.3 AdsSession Methods

The AdsSession [▶ 603] type exposes the following members.

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88] (Inherited from Session [▶ 101].)
	Connect [▶ 110]	Connects the session. (Inherited from Session [▶ 101].)
	Disconnect [▶ 111]	Disconnects the session from the target. (Inherited from Session [▶ 101].)
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [▶ 101].)
	Dispose(Boolean) [▶ 622]	Releases unmanaged and - optionally - managed resources. (Inherited from AdsSessionBase [▶ 614].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 623]	Finalizes an instance of the AdsSessionBase [▶ 614] class. (Inherited from AdsSessionBase [▶ 614].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 623]	Gets the name/string identifier of the session. (Inherited from AdsSessionBase [▶ 614].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 624]	Handler function connecting the Session. (Inherited from AdsSessionBase [▶ 614].)
	OnCreateSymbolServer [▶ 624]	Handler function creating the symbol server object. (Inherited from AdsSessionBase [▶ 614].)
	OnDisconnect [▶ 625]	Called when [disconnect]. (Inherited from AdsSessionBase [▶ 614].)

	Name	Description
	OnGetAddress [▶ 625]	Handler function getting the address of the session. (Inherited from AdsSessionBase [▶ 614].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsSession Class [▶ 603]

TwinCAT.Ads Namespace [▶ 151]

6.2.15.4 AdsSession Events

The AdsSession [▶ 603] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from Session [▶ 101].)

Reference

AdsSession Class [▶ 603]

TwinCAT.Ads Namespace [▶ 151]

6.2.16 AdsSessionBase Class

Abstract base class for ADS Sessions.

Inheritance Hierarchy

System.Object
  TwinCAT.Session [▶ 101]
    TwinCAT.Ads.AdsSessionBase
      TwinCAT.Ads.AdsSession [▶ 603]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class AdsSessionBase : Session,
    IAdsSession, ISession, IConnectionStateProvider, ISymbolServerProvider
```

The AdsSessionBase type exposes the following members.

Constructors

	Name	Description
	AdsSessionBase [▶ 616]	Initializes a new instance of the AdsSessionBase class.

Properties

	Name	Description
	Address [▶ 618]	Gets the target address of the AdsSessionBase
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation. (Inherited from Session [▶ 101].)
	Connection [▶ 618]	Gets the connection.
 	ConnectionState [▶ 104]	Gets the current Connection state of the Session [▶ 101] (Inherited from Session [▶ 101].)
	Disposed [▶ 105]	Gets a value indicating whether this Session [▶ 101] is disposed. (Inherited from Session [▶ 101].)
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established. (Inherited from Session [▶ 101].)
	Id [▶ 106]	Gets the Session Identifier (Inherited from Session [▶ 101].)
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected. (Inherited from Session [▶ 101].)
	Name [▶ 107]	Gets the name of the session (Inherited from Session [▶ 101].)
	NetId [▶ 619]	Gets the NetId of the Session
	Owner [▶ 619]	Gets the Session owner.
	Port [▶ 620]	Gets the Ams Port of the Session
	Settings [▶ 620]	Gets the settings of the connection.
	Statistics [▶ 621]	Gets the Communication / Session statistics.
	SymbolServer [▶ 108]	Gets the symbol server. (Inherited from Session [▶ 101].)

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88] (Inherited from Session [▶ 101].)
	Connect [▶ 110]	Connects the session. (Inherited from Session [▶ 101].)
	Disconnect [▶ 111]	Disconnects the session from the target. (Inherited from Session [▶ 101].)
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [▶ 101].)
	Dispose(Boolean) [▶ 622]	Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean) [▶ 112].)

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 623]	Finalizes an instance of the AdsSessionBase class. (Overrides Object.Finalize..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 623]	Gets the name/string identifier of the session. (Overrides Session.GetSessionName. [▶ 112].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 624]	Handler function connecting the Session. (Overrides Session.OnConnect(Boolean) [▶ 113].)
	OnCreateSymbolServer [▶ 624]	Handler function creating the symbol server object. (Overrides Session.OnCreateSymbolServer. [▶ 113].)
	OnDisconnect [▶ 625]	Called when [disconnect]. (Overrides Session.OnDisconnect. [▶ 114].)
	OnGetAddress [▶ 625]	Handler function getting the address of the session. (Overrides Session.OnGetAddress. [▶ 114].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from Session [▶ 101].)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.16.1 AdsSessionBase Constructor

Initializes a new instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsSessionBase(
    AmsAddress address,
    SessionSettings settings,
    IAdsClientFactory factory,
    Object owner
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.
settings	Type: TwinCAT.Ads.SessionSettings [▶ 1035] The settings.
factory	Type: IAdsClientFactory The client factory
owner	Type: System.Object The session owner

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2 AdsSessionBase Properties

The AdsSessionBase [▶ 614] type exposes the following members.

Properties

	Name	Description
	Address [▶ 618]	Gets the target address of the AdsSessionBase [▶ 614]
	AddressSpecifier [▶ 103]	Gets the communication endpoint address string representation. (Inherited from Session [▶ 101].)
	Connection [▶ 618]	Gets the connection.
 	ConnectionState [▶ 104]	Gets the current Connection state of the Session [▶ 101] (Inherited from Session [▶ 101].)
	Disposed [▶ 105]	Gets a value indicating whether this Session [▶ 101] is disposed. (Inherited from Session [▶ 101].)
	EstablishedAt [▶ 106]	Gets the UTC time when the session was established. (Inherited from Session [▶ 101].)
	Id [▶ 106]	Gets the Session Identifier (Inherited from Session [▶ 101].)
	IsConnected [▶ 107]	Gets a value indicating whether this instance is connected. (Inherited from Session [▶ 101].)
	Name [▶ 107]	Gets the name of the session (Inherited from Session [▶ 101].)
	NetId [▶ 619]	Gets the NetId of the Session
	Owner [▶ 619]	Gets the Session owner.
	Port [▶ 620]	Gets the Ams Port of the Session
	Settings [▶ 620]	Gets the settings of the connection.
	Statistics [▶ 621]	Gets the Communication / Session statistics.

	Name	Description
	SymbolServer [▶ 108]	Gets the symbol server. (Inherited from Session [▶ 101].)

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.1 AdsSessionBase.Address Property

Gets the target address of the AdsSessionBase [▶ 614]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The address.

Implements

IAdsSession.Address [▶ 910]

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.2 AdsSessionBase.Connection Property

Gets the connection.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsConnection Connection { get; protected set; }
```

Property Value

Type: AdsConnection [▶ 357]
The connection.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.3 AdsSessionBase.NetId Property

Gets the NetId of the Session

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The net identifier.

Implements

IAdsSession.NetId [▶ 911]

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.4 AdsSessionBase.Owner Property

Gets the Session owner.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object Owner { get; }
```

Property Value

Type: Object
The owner or NULL

Implements

IAdsSession.Owner [▶ 911]

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.5 AdsSessionBase.Port Property

Gets the Ams Port of the Session

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Port { get; }
```

Property Value

Type: Int32
The port.

Implements

IAdsSession.Port [▶ 911]

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.6 AdsSessionBase.Settings Property

Gets the settings of the connection.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SessionSettings Settings { get; }
```

Property Value

Type: SessionSettings [▶ 1035]
The settings.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.2.7 AdsSessionBase.Statistics Property

Gets the Communication / Session statistics.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsCommunicationStatistics Statistics { get; }
```

Property Value

Type: AdsCommunicationStatistics [▶ 350]
The communication / Session statistics.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3 AdsSessionBase Methods

The AdsSessionBase [▶ 614] type exposes the following members.

Methods

	Name	Description
	Close [▶ 109]	Closes this ISession [▶ 88] (Inherited from Session [▶ 101].)
	Connect [▶ 110]	Connects the session. (Inherited from Session [▶ 101].)
	Disconnect [▶ 111]	Disconnects the session from the target. (Inherited from Session [▶ 101].)
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [▶ 101].)
	Dispose(Boolean) [▶ 622]	Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean) [▶ 112].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 623]	Finalizes an instance of the AdsSessionBase [▶ 614] class. (Overrides Object.Finalize..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetSessionName [▶ 623]	Gets the name/string identifier of the session. (Overrides Session.GetSessionName. [▶ 112].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnConnect [▶ 624]	Handler function connecting the Session. (Overrides Session.OnConnect(Boolean) [▶ 113].)

	Name	Description
	OnCreateSymbolServer [▶ 624]	Handler function creating the symbol server object. (Overrides Session.OnCreateSymbolServer. [▶ 113].)
	OnDisconnect [▶ 625]	Called when [disconnect]. (Overrides Session.OnDisconnect. [▶ 114].)
	OnGetAddress [▶ 625]	Handler function getting the address of the session. (Overrides Session.OnGetAddress. [▶ 114].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.1 AdsSessionBase.Dispose Method

Overload List

	Name	Description
	Dispose. [▶ 112]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from Session [▶ 101].)
	Dispose(Boolean) [▶ 622]	Releases unmanaged and - optionally - managed resources. (Overrides Session.Dispose(Boolean) [▶ 112].)

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

AdsSessionBase.Dispose Method (Boolean)

Releases unmanaged and - optionally - managed resources.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override void Dispose(
    bool disposing
)
```

Parameters

disposing	Type: System.Boolean true to release both managed and unmanaged resources; false to release only unmanaged resources.

Reference

AdsSessionBase Class [▶ 614]

Dispose Overload [▶ 622]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.2 AdsSessionBase.Finalize Method

Finalizes an instance of the AdsSessionBase [▶ 614] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override void Finalize()
```

Implements

Object.Finalize.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.3 AdsSessionBase.GetSessionName Method

Gets the name/string identifier of the session.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override string GetSessionName()
```

Return Value

Type: String
System.String.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.4 AdsSessionBase.OnConnect Method

Handler function connecting the Session.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override IConnection OnConnect(
    bool reconnect
)
```

Parameters

reconnect	Type: System.Boolean

Return Value

Type: IConnection [▶ 74]
IConnection.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.5 AdsSessionBase.OnCreateSymbolServer Method

Handler function creating the symbol server object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override ISymbolServer OnCreateSymbolServer()
```

Return Value

Type: ISymbolServer [▶ 2205]
ISymbolServer.

Exceptions

Exception	Condition
SessionNotConnectedException [▶ 129]	The connection is not established!

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.6 AdsSessionBase.OnDisconnect Method

Called when [disconnect].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override bool OnDisconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.3.7 AdsSessionBase.OnGetAddress Method

Handler function getting the address of the session.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override string OnGetAddress()
```

Return Value

Type: String
System.String.

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.16.4 AdsSessionBase Events

The AdsSessionBase [▶ 614] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 115]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from Session [▶ 101].)

Reference

AdsSessionBase Class [▶ 614]

TwinCAT.Ads Namespace [▶ 151]

6.2.17 AdsState Enumeration

Describes the AdsState.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AdsState
```

Members

	Member name	Value	Description
	Invalid	0	Ads State is Invalid / Uninitialized
	Idle	1	Idle
	Reset	2	Reset
	Init	3	Initialize
	Start	4	Start
	Run	5	Run
	Stop	6	Stop
	SaveConfig	7	Save Configuration
	LoadConfig	8	Load Configuration
	PowerFailure	9	Power failure
	PowerGood	10	Power Good
	Error	11	Error
	Shutdown	12	Shutdown
	Suspend	13	Suspend
	Resume	14	Resume
	Config	15	Config (System is in config mode)
	Reconfig	16	Reconfig (System should restart in config mode)
	Stopping	17	Stopping
	Incompatible	18	Incompatible
	Exception	19	Exception

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.18 AdsStateChangedEventArgs Class

Arguments for the AdsStateChanged [▶ 934] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsStateChangedEventArgs
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsStateChangedEventArgs : EventArgs
```

The AdsStateChangedEventArgs type exposes the following members.

Constructors

	Name	Description
	AdsStateChangedEventArgs [▶ 627]	Initializes a new instance of the AdsStateChangedEventArgs class.

Properties

	Name	Description
	State [▶ 628]	Current state of the ADS device.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.18.1 AdsStateChangedEventArgs Constructor

Initializes a new instance of the AdsStateChangedEventArgs class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsStateChangedEventArgs(
    StateInfo value
)
```

Parameters

value	Type: TwinCAT.Ads.StateInfo [▶ 1041] Current state of the ADS device.

Reference

AdsStateChangedEventArgs Class [▶ 627]

TwinCAT.Ads Namespace [▶ 151]

6.2.18.2 AdsStateChangedEventArgs Properties

The AdsStateChangedEventArgs [▶ 627] type exposes the following members.

Properties

	Name	Description
	State [▶ 628]	Current state of the ADS device.

Reference

AdsStateChangedEventArgs Class [▶ 627]

TwinCAT.Ads Namespace [▶ 151]

6.2.18.2.1 AdsStateChangedEventArgs.State Property

Current state of the ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo State { get; }
```

Property Value

Type: StateInfo [▶ 1041]
The state.

Reference

AdsStateChangedEventArgs Class [▶ 627]

TwinCAT.Ads Namespace [▶ 151]

6.2.18.3 AdsStateChangedEventArgs Methods

The AdsStateChangedEventArgs [▶ 627] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsStateChangedEventArgs Class [▶ 627]

TwinCAT.Ads Namespace [▶ 151]

6.2.19 AdsStateChangedEventArgs2 Class

Event Arguments for AdsStateChanged events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsStateChangedEventArgs2

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsStateChangedEventArgs2 : EventArgs
```

The AdsStateChangedEventArgs2 type exposes the following members.

Properties

	Name	Description
	Connection [▶ 630]	Gets the connection.
	NewState [▶ 631]	The new state
	OldState [▶ 631]	The old state

	Name	Description
	Session [▶ 632]	The session

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

System.EventArgs

6.2.19.1 AdsStateChangedEventArgs2 Properties

The AdsStateChangedEventArgs2 [▶ 629] type exposes the following members.

Properties

	Name	Description
	Connection [▶ 630]	Gets the connection.
	NewState [▶ 631]	The new state
	OldState [▶ 631]	The old state
	Session [▶ 632]	The session

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.19.1.1 AdsStateChangedEventArgs2.Connection Property

Gets the connection.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]
The connection.

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.19.1.2 AdsStateChangedEventArgs2.NewState Property

The new state

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo NewState { get; }
```

Property Value

Type: StateInfo [▶ 1041]

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.19.1.3 AdsStateChangedEventArgs2.OldState Property

The old state

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo OldState { get; }
```

Property Value

Type: StateInfo [▶ 1041]

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.19.1.4 AdsStateChangedEventArgs2.Session Property

The session

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISession Session { get; }
```

Property Value

Type: ISession [▶ 88]

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.19.2 AdsStateChangedEventArgs2 Methods

The AdsStateChangedEventArgs2 [▶ 629] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsStateChangedEventArgs2 Class [▶ 629]

TwinCAT.Ads Namespace [▶ 151]

6.2.20 AdsSumCommandException Class

The exception that is thrown when an ADS SumCommandBase error occurs.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.Ads.AdsErrorException [▶ 583]
        TwinCAT.Ads.AdsSumCommandException

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class AdsSumCommandException : AdsErrorException
```

The AdsSumCommandException type exposes the following members.

Constructors

	Name	Description
	AdsSumCommandException [▶ 634]	Initializes a new Instance of the AdsErrorException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 587]	Gets the error code of the Exception. (Inherited from AdsErrorException [▶ 583].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	SumCommand [▶ 635]	Gets the sum command.
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetObjectData [▶ 636]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides AdsErrorException.GetObjectData(SerializationInfo, StreamingContext) [▶ 590].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.20.1 AdsSumCommandException Constructor

Initializes a new Instance of the AdsErrorException class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsSumCommandException(
    string message,
    ISumCommand command
)
```

Parameters

message	Type: System.String The message.
command	Type: TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206] The command.

Reference

AdsSumCommandException Class [▶ 632]

TwinCAT.Ads Namespace [▶ 151]

6.2.20.2 AdsSumCommandException Properties

The AdsSumCommandException [▶ 632] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 587]	Gets the error code of the Exception. (Inherited from AdsErrorException [▶ 583].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)

	Name	Description
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	SumCommand [▶ 635]	Gets the sum command.
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

AdsSumCommandException Class [▶ 632]

TwinCAT.Ads Namespace [▶ 151]

6.2.20.2.1 AdsSumCommandException.SumCommand Property

Gets the sum command.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISumCommand SumCommand { get; }
```

Property Value

Type: ISumCommand [▶ 1206]
The sum command.

Reference

AdsSumCommandException Class [▶ 632]

TwinCAT.Ads Namespace [▶ 151]

6.2.20.3 AdsSumCommandException Methods

The AdsSumCommandException [▶ 632] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 636]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides AdsErrorException.GetObjectData(SerializationInfo, StreamingContext) [▶ 590].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AdsSumCommandException Class [▶ 632]

TwinCAT.Ads Namespace [▶ 151]

6.2.20.3.1 AdsSumCommandException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

AdsSumCommandException Class [▶ 632]

TwinCAT.Ads Namespace [▶ 151]

6.2.21 AdsSymbolVersionChangedEventArgs Class

Arguments for the AdsSymbolVersionChanged [▶ 937] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsSymbolVersionChangedEventArgs

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AdsSymbolVersionChangedEventArgs : EventArgs
```

The AdsSymbolVersionChangedEventArgs type exposes the following members.

Properties

	Name	Description
	SymbolVersion [▶ 638]	Current symbol version device.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.21.1 AdsSymbolVersionChangedEventArgs Properties

The AdsSymbolVersionChangedEventArgs [▶ 637] type exposes the following members.

Properties

	Name	Description
	SymbolVersion [▶ 638]	Current symbol version device.

Reference

AdsSymbolVersionChangedEventArgs Class [▶ 637]

TwinCAT.Ads Namespace [▶ 151]

6.2.21.1.1 AdsSymbolVersionChangedEventArgs.SymbolVersion Property

Current symbol version device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public short SymbolVersion { get; }
```

Property Value

Type: Int16
The symbol version.

Reference

AdsSymbolVersionChangedEventArgs Class [▶ 637]

TwinCAT.Ads Namespace [▶ 151]

6.2.21.2 AdsSymbolVersionChangedEventArgs Methods

The AdsSymbolVersionChangedEventArgs [▶ 637] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsSymbolVersionChangedEventArgs Class [▶ 637]

TwinCAT.Ads Namespace [▶ 151]

6.2.22 AdsTransMode Enumeration

ADS Transmission Mode for ADS Notifications.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AdsTransMode
```

Members

	Member name	Value	Description
	None	0	None / Uninitalized transport mode. No AdsNotification [▶ 866] event is fired.
	ClientCycle	1	Client triggered cyclic AdsNotification [▶ 866] event. The AdsNotification [▶ 866] event is fired cyclically triggered from the client side. Polling is used from the User Application to read values, before they are fired as Notifications. Client side triggering has the following consequences: • The realtime environment on the server side will be less stressed (expecially the mailbox queue). • Value requests are serialized one after another and are handled slower (synchronouly, not asynchronously) • Implicit synchronization of the events into the UI Thread.
	ClientOnChange	2	The AdsNotification [▶ 866] event is fired when data changes triggered by the client. The AdsNotification [▶ 866] event is fired on-change triggered from the client side. Polling is used from the User Application to read values, before they are fired as Notifications. Client side triggering has the following consequences: • The realtime environment on the server side will be less stressed (expecially the mailbox queue). • Value requests are serialized one after another and are handled slower (synchronouly, not asynchronously) • Implicit synchronization of the events into the UI Thread.

	Member name	Value	Description
	Cyclic	3	The AdsNotification [▶ 866] event is fired cyclically. The Notification will be registered on the ADS Server side for a cyclical trigger (dependant on time parameter) and is bound to the 'default' task of the addressed target. In case of the PLC target (e.g. Port 851) the default task is the first configured task. Each time the 'default' task has finished its cycle the realtime system will check for the expired cycle time and sends the AdsNotification [▶ 866] message on expiry. The used ContextMask for the 'default' task is 0. **Please be aware, that server side 'Change' notifications stress the realtime system and should be handled with care. Therefore, dependent of the cycle time of the task and the capabilities of the system only a limited set of Cyclic Notifications should be used!** A system limit for server side notification registrations is 1024.
	OnChange	4	On-Change AdsNotification [▶ 866] event. The Notification will be registered on the ADS Server side for an on-change and optional cyclical trigger (dependant on parameters) and is bound to the 'default' task of the addressed target. In case of the PLC target (e.g. Port 851) the default task is the first configured task. Each time this task has finished its cycle the realtime system will check for the changed value and an optional expired cycle time and sends the AdsNotification [▶ 866] message on change or expiry. The used ContextMask for the 'default' task is 0. **Please be aware, that server side 'OnChange' notifications stress the realtime system / the default task with value comparisons. Therefore, dependent of the cycle time of the task and the capabilities of the system a higher amount of notification registrations should be handled with care !** A system limit for server side notification registrations is 1024.

	Member name	Value	Description
	CyclicInContext	5	The AdsNotification [▶ 866] event is fired cyclically within the given task context. A Value of parameter is interpreted as task context number ContextMask [▶ 1388]. This can be important, if the notifications have to be synchronous with specific tasks, but should not be used in the default case. The Notification will be registered on the ADS Server side for a cyclical trigger (dependant on time parameter) and is bound to the task specified by the ContextMask of the addressed target. In case of the PLC target (e.g. Port 851) the ContextMask is the Index of the global TASKINFOARRAY - 1. Each time this task has finished its cycle the realtime system will check for the expired cycle time and sends the AdsNotification [▶ 866] message on expiry.
	OnChangeInContext	6	The AdsNotification [▶ 866] event is fired when the data changes within the given task context. A Value of parameter is interpreted as task context number ContextMask [▶ 1388]. This can be important, if the notifications have to be synchronously with specific tasks, but should not be used in the default case. The Notification will be registered on the ADS Server side for an on-change and optional cyclical trigger (dependant on parameters) and is bound to the task specified by the ContextMask of the addressed target. In case of the PLC target (e.g. Port 851) the ContextMask is the Index of the global TASKINFOARRAY - 1. Each time this task has finished its cycle the realtime system will check for the changed value and an optional expired cycle time and sends the AdsNotification [▶ 866] message on change or expiry. **Please be aware, that server side 'OnChange' notifications stress the realtime system / the default task with value comparisons. Therefore, dependent of the cycle time of the task and the capabilities of the system only a limited set of OnChange Notifications should be used!**

Remarks

The AdsTransMode configures the registration of the AdsNotification [▶ 866] at the server system and how the parameters of the AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844] are interpreted. The following general scenarios are addressed:

- Cyclic notifications.
- Notifications on value change.
- Server side and Client side notifications.
- Binding of notifications to specific tasks.

In the default case the OnChange or the Cyclic (Server cycle) should be used. All other modes are side cases for special purposes.

More about the AdsNotifications: ADS Notification concept [▶ 22].

Reference

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.AdsNotificationEx [▶ 868]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationEx Overload [▶ 850]

6.2.23 AdsVersion Class

The structure contains the version number, revision number and build number.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AdsVersion
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AdsVersion
```

The AdsVersion type exposes the following members.

Constructors

	Name	Description
	AdsVersion(.Byte.) [▶ 644]	Initializes a new instance of the AdsVersion class.
	AdsVersion(Int32, Int32, Int32) [▶ 644]	Initializes a new instance of the AdsVersion struct.

Properties

	Name	Description
	Build [▶ 645]	Gets or sets the build number.
S	Empty [▶ 646]	Get the Empty/Uninitialized Version (0,0,0)
	IsEmpty [▶ 646]	Gets a value indicating whether this instance is empty / uninitialized.
	Revision [▶ 647]	Gets or sets the revision number.
	Version [▶ 647]	Gets or sets the version number.

Methods

	Name	Description
	ConvertToStandard [▶ 648]	Converts this AdsVersion to a .NET Framework Version [▶ 647] object.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.23.1 AdsVersion Constructor

Overload List

	Name	Description
	AdsVersion(.Byte.) [▶ 644]	Initializes a new instance of the AdsVersion [▶ 642] class.
	AdsVersion(Int32, Int32, Int32) [▶ 644]	Initializes a new instance of the AdsVersion [▶ 642] struct.

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.1.1 AdsVersion Constructor (.Byte.)

Initializes a new instance of the AdsVersion [▶ 642] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsVersion(
    byte[] bytes
)
```

Parameters

bytes	Type: .System.Byte. The bytes.

Exceptions

Exception	Condition
ArgumentNullException	bytes
ArgumentException	bytes

Reference

AdsVersion Class [▶ 642]

AdsVersion Overload [▶ 643]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.1.2 AdsVersion Constructor (Int32, Int32, Int32)

Initializes a new instance of the AdsVersion [▶ 642] struct.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsVersion(
    int version,
    int revision,
    int build
)
```

Parameters

version	Type: System.Int32 The version.
revision	Type: System.Int32 The revision.

build	Type: System.Int32 The build.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	version or revision

Reference

AdsVersion Class [▶ 642]

AdsVersion Overload [▶ 643]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2 AdsVersion Properties

The AdsVersion [▶ 642] type exposes the following members.

Properties

	Name	Description
	Build [▶ 645]	Gets or sets the build number.
	Empty [▶ 646]	Get the Empty/Uninitialized Version (0,0,0)
	IsEmpty [▶ 646]	Gets a value indicating whether this instance is empty / uninitialized.
	Revision [▶ 647]	Gets or sets the revision number.
	Version [▶ 647]	Gets or sets the version number.

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2.1 AdsVersion.Build Property

Gets or sets the build number.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Build { get; set; }
```

Property Value

Type: Int32

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2.2 AdsVersion.Empty Property

Get the Empty/Uninitialized Version (0,0,0)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AdsVersion Empty { get; }
```

Property Value

Type: AdsVersion [▶ 642]
The empty.

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2.3 AdsVersion.IsEmpty Property

Gets a value indicating whether this instance is empty / uninitialized.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsEmpty { get; }
```

Property Value

Type: Boolean
true if this instance is empty; otherwise, false.

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2.4 AdsVersion.Revision Property

Gets or sets the revision number.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte Revision { get; set; }
```

Property Value

Type: Byte

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.2.5 AdsVersion.Version Property

Gets or sets the version number.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte Version { get; set; }
```

Property Value

Type: Byte

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.3 AdsVersion Methods

The AdsVersion [▶ 642] type exposes the following members.

Methods

	Name	Description
	ConvertToStandard [▶ 648]	Converts this AdsVersion [▶ 642] to a .NET Framework Version [▶ 647] object.

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.23.3.1 AdsVersion.ConvertToStandard Method

Converts this AdsVersion [▶ 642] to a .NET Framework Version [▶ 647] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Version ConvertToStandard()
```

Return Value

Type: Version
Version.

Reference

AdsVersion Class [▶ 642]

TwinCAT.Ads Namespace [▶ 151]

6.2.24 AmsAddress Class

Ams/Ads Address

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AmsAddress
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AmsAddress
```

The AmsAddress type exposes the following members.

Constructors

	Name	Description
	AmsAddress. [▶ 651]	Protected constructor
	AmsAddress(Int32) [▶ 651]	Constructor
	AmsAddress(String) [▶ 652]	Initializes a new instance of the AmsAddress class.
	AmsAddress(AmsAddress) [▶ 652]	Copy constructor
	AmsAddress(AmsPort) [▶ 653]	Constructor
	AmsAddress(.Byte., Int32) [▶ 653]	Constructor
	AmsAddress(.Byte., AmsPort) [▶ 654]	Constructor
	AmsAddress(String, Int32) [▶ 655]	Constructor
	AmsAddress(String, AmsPort) [▶ 655]	Constructor
	AmsAddress(AmsNetId, Int32) [▶ 656]	Constructor
	AmsAddress(AmsNetId, AmsPort) [▶ 656]	Constructor

Properties

	Name	Description
 	Empty [▶ 657]	Gets an Empty Address.
	NetId [▶ 657]	Gets the NetId
	Port [▶ 658]	Gets the Port number

Methods

	Name	Description
	Clone [▶ 659]	Clones this instance.

	Name	Description
	Equals [▶ 659]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 660]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	Parse [▶ 660]	Parses a string to an AmsAddress object.
	SetNetId [▶ 661]	Sets the net identifier.
	SetPort [▶ 662]	Sets the port.
	ToString [▶ 662]	Converts the Address to String 'NetId:Port' (Overrides Object.ToString..)
S	TryParse [▶ 662]	Tries to parse the AmsAddress from string.

Operators

	Name	Description
S	Equality [▶ 663]	Operator==
S	Inequality [▶ 664]	Implements the != operator.

Remarks

The AmsAddress consists of NetId [▶ 657] and Port [▶ 658] information and once it is constructed is immutable.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1 AmsAddress Constructor

Overload List

	Name	Description
	AmsAddress. [▶ 651]	Protected constructor
	AmsAddress(Int32) [▶ 651]	Constructor
	AmsAddress(String) [▶ 652]	Initializes a new instance of the AmsAddress [▶ 648] class.

	Name	Description
	AmsAddress(AmsAddress) [▶ 652]	Copy constructor
	AmsAddress(AmsPort) [▶ 653]	Constructor
	AmsAddress(.Byte., Int32) [▶ 653]	Constructor
	AmsAddress(.Byte., AmsPort) [▶ 654]	Constructor
	AmsAddress(String, Int32) [▶ 655]	Constructor
	AmsAddress(String, AmsPort) [▶ 655]	Constructor
	AmsAddress(AmsNetId, Int32) [▶ 656]	Constructor
	AmsAddress(AmsNetId, AmsPort) [▶ 656]	Constructor

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.1 AmsAddress Constructor

Protected constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AmsAddress()
```

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.2 AmsAddress Constructor (Int32)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    int port
)
```

Parameters

port	Type: System.Int32 The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.3 AmsAddress Constructor (String)

Initializes a new instance of the AmsAddress [▶ 648] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    string str
)
```

Parameters

str	Type: System.String The address coded as string (Format NetId:Port, 1.2.3.4.5.6:Port)

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.4 AmsAddress Constructor (AmsAddress)

Copy constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    AmsAddress address
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.5 AmsAddress Constructor (AmsPort)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    AmsPort port
)
```

Parameters

port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.6 AmsAddress Constructor (.Byte., Int32)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    byte[] netId,
    int port
)
```

Parameters

netId	Type: .System.Byte. The net identifier.
port	Type: System.Int32 The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.7 AmsAddress Constructor (.Byte., AmsPort)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    byte[] netId,
    AmsPort port
)
```

Parameters

netId	Type: .System.Byte. The net identifier.
port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.8 AmsAddress Constructor (String, Int32)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    string netId,
    int port
)
```

Parameters

netId	Type: System.String The net identifier.
port	Type: System.Int32 The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.9 AmsAddress Constructor (String, AmsPort)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    string netId,
    AmsPort port
)
```

Parameters

netId	Type: System.String The net identifier.
port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.10 AmsAddress Constructor (AmsNetId, Int32)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    AmsNetId netId,
    int port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] Net Id
port	Type: System.Int32 Port

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.1.11 AmsAddress Constructor (AmsNetId, AmsPort)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress(
    AmsNetId netId,
    AmsPort port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net identifier.
port	Type: TwinCAT.Ads.AmsPort [▶ 693] The port.

Reference

AmsAddress Class [▶ 648]

AmsAddress Overload [▶ 650]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.2 AmsAddress Properties

The AmsAddress [▶ 648] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 657]	Gets an Empty Address.
	NetId [▶ 657]	Gets the NetId
	Port [▶ 658]	Gets the Port number

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.2.1 AmsAddress.Empty Property

Gets an Empty Address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsAddress Empty { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The empty.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.2.2 AmsAddress.NetId Property

Gets the NetId

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The net identifier.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.2.3 AmsAddress.Port Property

Gets the Port number

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Port { get; }
```

Property Value

Type: Int32
The port.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3 AmsAddress Methods

The AmsAddress [▶ 648] type exposes the following members.

Methods

	Name	Description
	Clone [▶ 659]	Clones this instance.
	Equals [▶ 659]	Equals (Overrides Object.Equals(Object).)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 660]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	Parse [▶ 660]	Parses a string to an AmsAddress [▶ 648] object.
	SetNetId [▶ 661]	Sets the net identifier.
	SetPort [▶ 662]	Sets the port.
	ToString [▶ 662]	Converts the Address to String 'NetId:Port' (Overrides Object.ToString..)
S	TryParse [▶ 662]	Tries to parse the AmsAddress [▶ 648] from string.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.1 AmsAddress.Clone Method

Clones this instance.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress Clone()
```

Return Value

Type: AmsAddress [▶ 648]
AmsAddress.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.2 AmsAddress.Equals Method

Equals

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.3 AmsAddress.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.4 AmsAddress.Parse Method

Parses a string to an AmsAddress [▶ 648] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsAddress Parse(
    string str
)
```

Parameters

str	Type: System.String The string.

Return Value

Type: AmsAddress [▶ 648]
AmsAddress.

Exceptions

Exception	Condition
FormatException	

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.5 AmsAddress.SetNetId Method

Sets the net identifier.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetNetId(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net identifier.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.6 AmsAddress.SetPort Method

Sets the port.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetPort(
    int port
)
```

Parameters

port	Type: System.Int32 The port.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.7 AmsAddress.ToString Method

Converts the Address to String 'NetId:Port'

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.3.8 AmsAddress.TryParse Method

Tries to parse the AmsAddress [▶ 648] from string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string str,
    out AmsAddress address
)
```

Parameters

str	Type: System.String The STR.
address	Type: TwinCAT.Ads.AmsAddress [▶ 648]. The address.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.4 AmsAddress Operators

The AmsAddress [▶ 648] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 663]	Operator==
	Inequality [▶ 664]	Implements the != operator.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.4.1 AmsAddress.Equality Operator

Operator==

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    AmsAddress o1,
    AmsAddress o2
)
```

Parameters

o1	Type: TwinCAT.Ads.AmsAddress [▶ 648] The o1.
o2	Type: TwinCAT.Ads.AmsAddress [▶ 648] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.24.4.2 AmsAddress.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    AmsAddress o1,
    AmsAddress o2
)
```

Parameters

o1	Type: TwinCAT.Ads.AmsAddress [▶ 648] The o1.
o2	Type: TwinCAT.Ads.AmsAddress [▶ 648] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

AmsAddress Class [▶ 648]

TwinCAT.Ads Namespace [▶ 151]

6.2.25 AmsNetId Class

AMS/ADS Net ID

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.AmsNetId
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class AmsNetId : IComparable<AmsNetId>,
    IComparable
```

The AmsNetId type exposes the following members.

Constructors

	Name	Description
	AmsNetId(.Byte.) [▶ 668]	Constructor
	AmsNetId(ReadOnlySpan) [▶ 668]	
	AmsNetId(String) [▶ 669]	Constructor
	AmsNetId(AmsNetId) [▶ 669]	Copy Constructor

Properties

	Name	Description
S	Broadcast [▶ 670]	Gets the broadcast address (255.255.255.255.255.255)
S	Empty [▶ 671]	Creates an empty NetId ("0.0.0.0.0.0")
	IsBroadcast [▶ 671]	Gets a value indicating whether this AmsNetId is the broadcast address (255.255.255.255.255.255)
	IsEmpty [▶ 672]	Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)
	IsLocal [▶ 672]	Determines, whether the AmsNetId is local.
	IsLoopback [▶ 673]	Indicates, that this AmsNetId indicates a Loopback (ID: 127.0.0.1.1.1)
	IsSubAddress [▶ 673]	Gets a value indicating whether this AmsNetId indicates a SubNet.

	Name	Description
	Item [▶ 674]	Gets or sets the Byte with the specified index.
S	Local [▶ 674]	Gets the Local Net ID (System service must be running)
S	LocalHost [▶ 675]	Creates the local NetId ("127.0.0.1.1.1")

Methods

	Name	Description
	AsSpan [▶ 676]	Returns the AmsNetId as byte Span.
	Clone [▶ 677]	Clones the NetId
	CompareTo(Object) [▶ 678]	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.
	CompareTo(AmsNetId) [▶ 678]	Compares the current object with another object of the same type.
	Equals(Object) [▶ 679]	Equals (Overrides Object.Equals(Object).)
S	Equals(AmsNetId, AmsNetId) [▶ 680]	Determines whether the specified AmsNetIds are equal.
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
S	FromBinHexString [▶ 681]	Creates the AmsNetId from bin hex string.
	GetHashCode [▶ 681]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	IsSameTarget(AmsNetId, AmsNetId) [▶ 682]	Determines whether the AmsNetIds refer to the same target.
S	IsSameTarget(AmsNetId, AmsNetId, Boolean) [▶ 683]	Determines whether the AmsNetIds refer to the same target.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	NetIdsEqual(.Byte.) [▶ 684]	Compares the netIds
S	NetIdsEqual(.Byte., .Byte.) [▶ 685]	Compares the NetIds

	Name	Description
	NetIdsEqual(.Byte., .Byte., Boolean) [▶ 685]	Compares the NetIds for SubID equality.
	Parse [▶ 686]	Converts the string representation of the address to AmsNetId.
	ToBinHex. [▶ 687]	Converts the AmsNetId to a BinHex string.
	ToBinHex(AmsNetId) [▶ 687]	Converts the specified AmsNetId to a BinHex string.
	ToBytes [▶ 688]	Converts the NetId object to byte array
	ToString. [▶ 689]	Converts the netId to string (Overrides Object.ToString.)
	ToString(String, IFormatProvider) [▶ 689]	Returns a String that represents this instance.
	TryParse [▶ 690]	Converts the string representation of the address to AmsNetId.
	TryWriteBytes [▶ 691]	

Operators

	Name	Description
	Equality [▶ 692]	Operator==
	Inequality [▶ 692]	Implements the != operator.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.25.1 AmsNetId Constructor

Overload List

	Name	Description
	AmsNetId(.Byte.) [▶ 668]	Constructor
	AmsNetId(ReadOnlySpan) [▶ 668]	
	AmsNetId(String) [▶ 669]	Constructor

	Name	Description
	AmsNetId(AmsNetId) [▶ 669]	Copy Constructor

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.1.1 AmsNetId Constructor (.Byte.)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId(
    byte[] netId
)
```

Parameters

netId	Type: .System.Byte. Net ID in bytes

Exceptions

Exception	Condition
ArgumentException	Not a valid NetId;netId

Reference

AmsNetId Class [▶ 665]

AmsNetId Overload [▶ 667]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.1.2 AmsNetId Constructor (ReadOnlySpan`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId(
    ReadOnlySpan data
)
```

Parameters

data	Type: ReadOnlySpan

Reference

AmsNetId Class [▶ 665]

AmsNetId Overload [▶ 667]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.1.3 AmsNetId Constructor (String)

Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId(
    string netId
)
```

Parameters

netId	Type: System.String NetID as string

Reference

AmsNetId Class [▶ 665]

AmsNetId Overload [▶ 667]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.1.4 AmsNetId Constructor (AmsNetId)

Copy Constructor

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] Net Id.

Reference

AmsNetId Class [▶ 665]

AmsNetId Overload [▶ 667]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2 AmsNetId Properties

The AmsNetId [▶ 665] type exposes the following members.

Properties

	Name	Description
	Broadcast [▶ 670]	Gets the broadcast address (255.255.255.255.255.255)
	Empty [▶ 671]	Creates an empty NetId ("0.0.0.0.0.0")
	IsBroadcast [▶ 671]	Gets a value indicating whether this AmsNetId [▶ 665] is the broadcast address (255.255.255.255.255.255)
	IsEmpty [▶ 672]	Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)
	IsLocal [▶ 672]	Determines, whether the AmsNetId [▶ 665] is local.
	IsLoopback [▶ 673]	Indicates, that this AmsNetId [▶ 665] indicates a Loopback (ID: 127.0.0.1.1.1)
	IsSubAddress [▶ 673]	Gets a value indicating whether this AmsNetId [▶ 665] indicates a SubNet.
	Item [▶ 674]	Gets or sets the Byte with the specified index.
	Local [▶ 674]	Gets the Local Net ID (System service must be running)
	LocalHost [▶ 675]	Creates the local NetId ("127.0.0.1.1.1")

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.1 AmsNetId.Broadcast Property

Gets the broadcast address (255.255.255.255.255.255)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId Broadcast { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The broadcast.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.2 AmsNetId.Empty Property

Creates an empty NetId ("0.0.0.0.0.0")

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId Empty { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The empty.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.3 AmsNetId.IsBroadcast Property

Gets a value indicating whether this AmsNetId [▶ 665] is the broadcast address (255.255.255.255.255.255)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsBroadcast { get; }
```

Property Value

Type: Boolean
true if this instance is broadcast; otherwise, false.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.4 AmsNetId.IsEmpty Property

Gets a value indicating whether this instance is empty / Uninitialized (AmsNetId: 0.0.0.0.0.0)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsEmpty { get; }
```

Property Value

Type: Boolean
true if this instance is empty; otherwise, false.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.5 AmsNetId.IsLocal Property

Determines, whether the AmsNetId [▶ 665] is local.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsLocal { get; }
```

Property Value

Type: Boolean
true if this instance is local; otherwise, false.

Remarks

Doesn't check for LocalHost [▶ 675].

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.6 AmsNetId.IsLoopback Property

Indicates, that this AmsNetId [▶ 665] indicates a Loopback (ID: 127.0.0.1.1.1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsLoopback { get; }
```

Property Value

Type: Boolean
true if this instance is loop back; otherwise, false.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.7 AmsNetId.IsSubAddress Property

Gets a value indicating whether this AmsNetId [▶ 665] indicates a SubNet.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsSubAddress { get; }
```

Property Value

Type: Boolean
true if this instance indicates a subnet; otherwise, false.

Remarks

A Subnet is indicated, when the AmsNetId [▶ 665] doesn't end with **X.X.X.X.1.1**

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.8 AmsNetId.Item Property

Gets or sets the Byte with the specified index.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte this[
    int i
] { get; set; }
```

Parameters

i		Type: System.Int32 The index.

Return Value

Type: Byte
System.Byte.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.9 AmsNetId.Local Property

Gets the Local Net ID (System service must be running)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId Local { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The local.

Remarks

The system service must be running

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.2.10 AmsNetId.LocalHost Property

Creates the local NetId ("127.0.0.1.1.1")

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId LocalHost { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The local host.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3 AmsNetId Methods

The AmsNetId [▶ 665] type exposes the following members.

Methods

	Name	Description
	AsSpan [▶ 676]	Returns the AmsNetId [▶ 665] as byte Span.
	Clone [▶ 677]	Clones the NetId
	CompareTo(Object) [▶ 678]	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.
	CompareTo(AmsNetId) [▶ 678]	Compares the current object with another object of the same type.
	Equals(Object) [▶ 679]	Equals (Overrides Object.Equals(Object).)
S	Equals(AmsNetId, AmsNetId) [▶ 680]	Determines whether the specified AmsNetId [▶ 665]s are equal.
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
S	FromBinHexString [▶ 681]	Creates the AmsNetId [▶ 665] from bin hex string.
	GetHashCode [▶ 681]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	IsSameTarget(AmsNetId, AmsNetId) [▶ 682]	Determines whether the AmsNetId [▶ 665]s refer to the same target.
	IsSameTarget(AmsNetId, AmsNetId, Boolean) [▶ 683]	Determines whether the AmsNetId [▶ 665]s refer to the same target.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	NetIdsEqual(.Byte.) [▶ 684]	Compares the netIds
	NetIdsEqual(.Byte., .Byte.) [▶ 685]	Compares the NetIds
	NetIdsEqual(.Byte., .Byte., Boolean) [▶ 685]	Compares the NetIds for SubID equality.
	Parse [▶ 686]	Converts the string representation of the address to AmsNetId [▶ 665].
	ToBinHex. [▶ 687]	Converts the AmsNetId [▶ 665] to a BinHex string.
	ToBinHex(AmsNetId) [▶ 687]	Converts the specified AmsNetId [▶ 665] to a BinHex string.
	ToBytes [▶ 688]	Converts the NetId object to byte array
	ToString. [▶ 689]	Converts the netId to string (Overrides Object.ToString.)
	ToString(String, IFormatProvider) [▶ 689]	Returns a String that represents this instance.
	TryParse [▶ 690]	Converts the string representation of the address to AmsNetId [▶ 665].
	TryWriteBytes [▶ 691]	

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.1 AmsNetId.AsSpan Method

Returns the AmsNetId [▶ 665] as byte Span.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlySpan AsSpan()
```

Return Value

Type: ReadOnlySpan
ReadOnlySpan<System.Byte>.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.2 AmsNetId.Clone Method

Clones the NetId

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId Clone()
```

Return Value

Type: AmsNetId [▶ 665]
The cloned AmsNetId [▶ 665]

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.3 AmsNetId.CompareTo Method

Overload List

	Name	Description
	CompareTo(Object) [▶ 678]	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.
	CompareTo(AmsNetId) [▶ 678]	Compares the current object with another object of the same type.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.CompareTo Method (Object)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int CompareTo(
    Object obj
)
```

Parameters

obj	Type: System.Object An object to compare with this instance.

Return Value

Type: Int32
A value that indicates the relative order of the objects being compared. The return value has these meanings: Value Meaning Less than zero This instance precedes obj in the sort order. Zero This instance occurs in the same position in the sort order as obj. Greater than zero This instance follows obj in the sort order.

Implements

IComparable.CompareTo(Object)

Reference

AmsNetId Class [▶ 665]

CompareTo Overload [▶ 677]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.CompareTo Method (AmsNetId)

Compares the current object with another object of the same type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int CompareTo(
    AmsNetId other
)
```

Parameters

other	Type: TwinCAT.Ads.AmsNetId [▶ 665] An object to compare with this object.

Return Value

Type: Int32
A value that indicates the relative order of the objects being compared. The return value has the following meanings: Value Meaning Less than zero This object is less than the other parameter.Zero This object is equal to other. Greater than zero This object is greater than other.

Implements

IComparable.T..CompareTo(T)

Reference

AmsNetId Class [▶ 665]

CompareTo Overload [▶ 677]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.4 AmsNetId.Equals Method

Overload List

	Name	Description
	Equals(Object) [▶ 679]	Equals (Overrides Object.Equals(Object).)
	Equals(AmsNetId, AmsNetId) [▶ 680]	Determines whether the specified AmsNetId [▶ 665]s are equal.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.Equals Method (Object)

Equals

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

AmsNetId Class [▶ 665]

Equals Overload [▶ 679]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.Equals Method (AmsNetId, AmsNetId)

Determines whether the specified AmsNetId [▶ 665]s are equal.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool Equals(
    AmsNetId netIDA,
    AmsNetId netIDB
)
```

Parameters

netIDA	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net IDA.
netIDB	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net IDB.

Return Value

Type: Boolean
true if the specified net IDA is equal; otherwise, false.

Reference

AmsNetId Class [▶ 665]

Equals Overload [▶ 679]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.5 AmsNetId.FromBinHexString Method

Creates the AmsNetId [▶ 665] from bin hex string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId FromBinHexString(
    string str
)
```

Parameters

str	Type: System.String The BinHex string.

Return Value

Type: AmsNetId [▶ 665]
AmsNetId.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.6 AmsNetId.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.7 AmsNetId.IsSameTarget Method

Overload List

	Name	Description
	IsSameTarget(AmsNetId, AmsNetId) [▶ 682]	Determines whether the AmsNetId [▶ 665]s refer to the same target.
	IsSameTarget(AmsNetId, AmsNetId, Boolean) [▶ 683]	Determines whether the AmsNetId [▶ 665]s refer to the same target.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.IsSameTarget Method (AmsNetId, AmsNetId)

Determines whether the AmsNetId [▶ 665]s refer to the same target.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool IsSameTarget(
    AmsNetId netIDA,
    AmsNetId netIDB
)
```

Parameters

netIDA	Type: TwinCAT.Ads.AmsNetId [▶ 665] NetID of target system A
netIDB	Type: TwinCAT.Ads.AmsNetId [▶ 665] NetID of target system B

Return Value

Type: Boolean
true if the target systems are the same, otherwise false.

Remarks

In comparison to the Equals(Object) [▶ 679] or Equals(AmsNetId, AmsNetId) [▶ 680] methods, this Method alse checks against the LocalHost ID, which means that LocalHost [▶ 675] is the same target as Local [▶ 674]

Reference

AmsNetId Class [▶ 665]

IsSameTarget Overload [▶ 682]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.Equals(Object) [▶ 679]

AmsNetId.Equals(AmsNetId, AmsNetId) [▶ 680]

AmsNetId.IsSameTarget Method (AmsNetId, AmsNetId, Boolean)

Determines whether the AmsNetId [▶ 665]s refer to the same target.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool IsSameTarget(
    AmsNetId netIDA,
    AmsNetId netIDB,
    bool ignoreSubId
)
```

Parameters

netIDA	Type: TwinCAT.Ads.AmsNetId [▶ 665] NetID of target system A
netIDB	Type: TwinCAT.Ads.AmsNetId [▶ 665] NetID of target system B
ignoreSubId	Type: System.Boolean Indicates only to check the 4 parts of the address (for SubId check).

Return Value

Type: Boolean
true if the target systems are the same, otherwise false.

Exceptions

Exception	Condition
ArgumentNullException	netIDA
ArgumentNullException	netIDB

Remarks

In comparison to the Equals(Object) [▶ 679] or Equals(AmsNetId, AmsNetId) [▶ 680] methods, this Method alse checks against the LocalHost ID, which means that LocalHost [▶ 675] is the same target as Local [▶ 674]

Reference

AmsNetId Class [▶ 665]

IsSameTarget Overload [▶ 682]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.Equals(Object) [▶ 679]

AmsNetId.Equals(AmsNetId, AmsNetId) [▶ 680]

6.2.25.3.8 AmsNetId.NetIdsEqual Method

Overload List

	Name	Description
	NetIdsEqual(.Byte.) [▶ 684]	Compares the netIds
S	NetIdsEqual(.Byte., .Byte.) [▶ 685]	Compares the NetIds
S	NetIdsEqual(.Byte., .Byte., Boolean) [▶ 685]	Compares the NetIds for SubID equality.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.NetIdsEqual Method (.Byte.)

Compares the netIds

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool NetIdsEqual(
    byte[] netId
)
```

Parameters

netId	Type: .System.Byte.	NetId in bytes.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

AmsNetId Class [▶ 665]

NetIdsEqual Overload [▶ 684]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.NetIdsEqual Method (.Byte., .Byte.)

Compares the NetIds

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool NetIdsEqual(
    byte[] netId1,
    byte[] netId2
)
```

Parameters

netId1	Type: .System.Byte. NetID1
netId2	Type: .System.Byte. NetId2

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

AmsNetId Class [▶ 665]

NetIdsEqual Overload [▶ 684]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.NetIdsEqual Method (.Byte., .Byte., Boolean)

Compares the NetIds for SubID equality.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool NetIdsEqual(
    byte[] netId1,
    byte[] netId2,
    bool ignoreSubID
)
```

Parameters

netId1	Type: .System.Byte. NetID1
netId2	Type: .System.Byte. NetId2
ignoreSubID	Type: System.Boolean Ignores the SubId part of the NetId (last 2 numbers/bytes).

Return Value

Type: Boolean
true if the SubIds are equal, false otherwise.

Remarks

The first 4 numbers indicate the global ID of the AmsNetId. The last 2 numbers indicate the (local) submodule.

Reference

AmsNetId Class [▶ 665]

NetIdsEqual Overload [▶ 684]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.9 AmsNetId.Parse Method

Converts the string representation of the address to AmsNetId [▶ 665].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId Parse(
    string str
)
```

Parameters

str	Type: System.String The string to parse.

Return Value

Type: AmsNetId [▶ 665]
AmsNetId.

Exceptions

Exception	Condition
FormatException	Format of AmsNetId is not valid!

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.10 AmsNetId.ToBinHex Method

Overload List

	Name	Description
	ToBinHex. [▶ 687]	Converts the AmsNetId [▶ 665] to a BinHex string.
S	ToBinHex(AmsNetId) [▶ 687]	Converts the specified AmsNetId [▶ 665] to a BinHex string.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.ToBinHex Method

Converts the AmsNetId [▶ 665] to a BinHex string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ToBinHex()
```

Return Value

Type: String
System.String.

Reference

AmsNetId Class [▶ 665]

ToBinHex Overload [▶ 687]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.ToBinHex Method (AmsNetId)

Converts the specified AmsNetId [▶ 665] to a BinHex string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static string ToBinHex(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] NetId to convert

Return Value

Type: String
System.String.

Reference

AmsNetId Class [▶ 665]

ToBinHex Overload [▶ 687]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.11 AmsNetId.ToBytes Method

Converts the NetId object to byte array

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] ToBytes()
```

Return Value

Type: .Byte.
System.Byte[].

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.12 AmsNetId.ToString Method

Overload List

	Name	Description
	ToString. [▶ 689]	Converts the netId to string (Overrides Object.ToString..)

	Name	Description
	ToString(String, IFormatProvider) [▶ 689]	Returns a String that represents this instance.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.ToString Method

Converts the netId to string

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

AmsNetId Class [▶ 665]

ToString Overload [▶ 688]

TwinCAT.Ads Namespace [▶ 151]

AmsNetId.ToString Method (String, IFormatProvider)

Returns a String that represents this instance.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ToString(
    string format,
    IFormatProvider formatProvider
)
```

Parameters

format	Type: System.String The format.

formatProvider	Type: System.IFormatProvider The format provider.

Return Value

Type: String
A String that represents this instance.

Exceptions

Exception	Condition
FormatException	

Remarks

Formatting	Description
g	Standard formatting
x	Formatting as Hexadecimal (small letters)
X	Formatting as Hexadecimal (big letters)

Reference

AmsNetId Class [▶ 665]

ToString Overload [▶ 688]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.13 AmsNetId.TryParse Method

Converts the string representation of the address to AmsNetId [▶ 665].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string str,
    out AmsNetId netId
)
```

Parameters

str	Type: System.String The string to parse.
netId	Type: TwinCAT.Ads.AmsNetId [▶ 665]. The parsed AmsNetId [▶ 665].

Return Value

Type: Boolean
true if parsed, false otherwise.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.3.14 AmsNetId.TryWriteBytes Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryWriteBytes(
    Span span
)
```

Parameters

span	Type: Span

Return Value

Type: Boolean

Reference

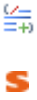
AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.4 AmsNetId Operators

The AmsNetId [▶ 665] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 692]	Operator==
	Inequality [▶ 692]	Implements the != operator.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.4.1 AmsNetId.Equality Operator

Operator==

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    AmsNetId o1,
    AmsNetId o2
)
```

Parameters

o1	Type: TwinCAT.Ads.AmsNetId [▶ 665] The o1.
o2	Type: TwinCAT.Ads.AmsNetId [▶ 665] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.25.4.2 AmsNetId.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    AmsNetId o1,
    AmsNetId o2
)
```

Parameters

o1	Type: TwinCAT.Ads.AmsNetId [▶ 665] The o1.
o2	Type: TwinCAT.Ads.AmsNetId [▶ 665] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

AmsNetId Class [▶ 665]

TwinCAT.Ads Namespace [▶ 151]

6.2.26 AmsPort Enumeration

AmsPorts

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AmsPort
```

Members

	Member name	Value	Description
	Router	1	AMS Router (Port 1)
	Debugger	2	AMS Debugger (Port 2)
	R0_TComServer	10	The TCom Server. Dpc or passive level.
	R0_TComServerTask	11	TCom Server Task. RT context.
	R0_TComServer_PL	12	TCom Serve Task. Passive level.
	R0_TcDebugger	20	TwinCAT Debugger
	R0_TcDebuggerTask	21	TwinCAT Debugger Task
	R0_LicenseServer	30	The License Server (Port 30)
	Logger	100	Logger (Port 100)
	EventLog	110	Event Logger (Port 110)
	DeviceApplication	120	application for coupler (EK), gateway (EL), etc.
	EventLog_UM	130	Event Logger UM
	EventLog_RT	131	Event Logger RT
	EventLogPublisher	132	Event Logger Publisher
	R0_Realtime	200	R0 Realtime (Port 200)
	R0_Trace	290	R0 Trace (Port 290)
	R0_IO	300	R0 IO (Port 300)
	R0_NC	500	NC (R0) (Port 500)
	R0_NCSAF	501	R0 Satzausführung (Port 501)
	R0_NCSVB	511	R0 Satzvorbereitung (Port 511)
	R0_NCINSTANCE	520	Preconfigured Nc2-Nc3-Instance
	R0_ISG	550	R0 ISG (Port 550)
	R0_CNC	600	R0 CNC (Port 600)
	R0_LINE	700	R0 Line (Port 700)

	Member name	Value	Description
	R0_PLC	800	R0 PLC (Port 800)
	Tc2_Plc1	801	Tc2 PLC RuntimeSystem 1 (Port 801)
	Tc2_Plc2	811	Tc2 PLC RuntimeSystem 2 (Port 811)
	Tc2_Plc3	821	Tc2 PLC RuntimeSystem 3 (Port 821)
	Tc2_Plc4	831	Tc2 PLC RuntimeSystem 4 (Port 831)
	R0_RTS	850	R0 RTS (Port 850)
	CamshaftController	900	Camshaft Controller (R0) (Port 900)
	R0_CAMTOOL	950	R0 CAM Tool (Port 950)
	R0_USER	2000	R0 User (Port 2000)
	SystemService	10000	System Service (AMSPORT_R3_SYSSERV, 10000)
	R3_CTRLPROG	10000	(Port 10000)
	R3_SYSCTRL	10001	(Port 10001)
	R3_SYSSAMPLER	10100	Port 10100
	R3_TCPRAWCONN	10200	Port 10200
	R3_TCPIPSERVER	10201	Port 10201
	R3_SYSMANAGER	10300	Port 10300
	R3_SMSSERVER	10400	Port 10400
	R3_MODBUSSERVER	10500	Port 10500
	R3_AMSLOGGER	10502	Port 10502
	R3_S7SERVER	10600	**Obsolete.** Port 10600
	R3_XMLDATASERVER	10600	Port 10600
	R3_AUTOCONFIG	10700	Port 10700
	R3_PLCCONTROL	10800	Port 10800
	R3_FTPCLIENT	10900	Port 10900
	R3_NCCTRL	11000	Port 11000
	R3_NCINTERPRETER	11500	Port 11500
	R3_GSTINTERPRETER	11600	Port 11600
	R3_STRECKECTRL	12000	Port 12000
	R3_CAMCTRL	13000	Port 13000
	R3_SCOPE	14000	Port 14000
	R3_CONDITIONMON	14100	Port 14100
	R3_SINECH1	15000	Port 15000
	R3_CONTROLNET	16000	Port 16000
	R3_OPCSERVER	17000	Port 17000
	R3_OPCCLIENT	17500	Port 17500
	R3_MAILSERVER	18000	Port 18000
	R3_EL60XX	19000	Port 19000
	R3_MANAGEMENT	19100	Port 19100
	R3_MIELEHOME	19200	Port 19200
	R3_CPLINK3	19300	Port 19300
	R3_VNSERVICE	19500	Port 19500
	R3_MULTIUSER	19600	Multiuser (Port 19600)
	USEDEFAULT	65535	

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.27 AmsRouterNotificationEventArgs Class

Arguments for the IRouterNotificationProvider [▶ 973] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AmsRouterNotificationEventArgs
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AmsRouterNotificationEventArgs : EventArgs
```

The AmsRouterNotificationEventArgs type exposes the following members.

Constructors

	Name	Description
	AmsRouterNotificationEventArgs [▶ 695]	Initializes a new instance of the AmsRouterNotificationEventArgs class.

Properties

	Name	Description
	State [▶ 696]	Current state of the AMS Router.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.27.1 AmsRouterNotificationEventArgs Constructor

Initializes a new instance of the AmsRouterNotificationEventArgs class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsRouterNotificationEventArgs(
    AmsRouterState value
)
```

Parameters

value	Type: TwinCAT.Ads.AmsRouterState [▶ 697] Value of the ADS variable.

Reference

AmsRouterNotificationEventArgs Class [▶ 695]

TwinCAT.Ads Namespace [▶ 151]

6.2.27.2 AmsRouterNotificationEventArgs Properties

The AmsRouterNotificationEventArgs [▶ 695] type exposes the following members.

Properties

	Name	Description
	State [▶ 696]	Current state of the AMS Router.

Reference

AmsRouterNotificationEventArgs Class [▶ 695]

TwinCAT.Ads Namespace [▶ 151]

6.2.27.2.1 AmsRouterNotificationEventArgs.State Property

Current state of the AMS Router.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsRouterState State { get; }
```

Property Value

Type: AmsRouterState [▶ 697]

Reference

AmsRouterNotificationEventArgs Class [▶ 695]

TwinCAT.Ads Namespace [▶ 151]

6.2.27.3 AmsRouterNotificationEventArgs Methods

The AmsRouterNotificationEventArgs [▶ 695] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AmsRouterNotificationEventArgs Class [▶ 695]

TwinCAT.Ads Namespace [▶ 151]

6.2.28 AmsRouterState Enumeration

State of the AMS Router.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AmsRouterState
```

Members

	Member name	Value	Description
	None	0	Unknown Router State (None, Uninitialized)
	Stop	1	AMS Router is stopped.
	Start	2	AMS Router is started.
	Removed	3	AMS Router has been removed.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.29 DeviceInfo Class

The structure contains the name and the version information of the device.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.DeviceInfo
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DeviceInfo
```

The DeviceInfo type exposes the following members.

Constructors

	Name	Description
	DeviceInfo [▶ 699]	Initializes a new instance of the DeviceInfo class.

Properties

	Name	Description
	Empty [▶ 699]	Empty / Uninitalized DeviceInfo
	IsEmpty [▶ 700]	Gets a value indicating whether this instance is empty.
	Name [▶ 700]	Gets the name of the device.
	Version [▶ 701]	Gets the version information.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.29.1 DeviceInfo Constructor

Initializes a new instance of the DeviceInfo [▶ 698] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DeviceInfo(
    string name,
    AdsVersion version
)
```

Parameters

name	Type: System.String The name.
version	Type: TwinCAT.Ads.AdsVersion [▶ 642] The version.

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.2 DeviceInfo Properties

The DeviceInfo [▶ 698] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 699]	Empty / Uninitalized DeviceInfo
	IsEmpty [▶ 700]	Gets a value indicating whether this instance is empty.
	Name [▶ 700]	Gets the name of the device.
	Version [▶ 701]	Gets the version information.

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.2.1 DeviceInfo.Empty Property

Empty / Uninitalized DeviceInfo

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DeviceInfo Empty { get; }
```

Property Value

Type: DeviceInfo [▶ 698]
The empty.

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.2.2 DeviceInfo.IsEmpty Property

Gets a value indicating whether this instance is empty.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsEmpty { get; }
```

Property Value

Type: Boolean
true if this instance is empty; otherwise, false.

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.2.3 DeviceInfo.Name Property

Gets the name of the device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.2.4 DeviceInfo.Version Property

Gets the version information.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsVersion Version { get; }
```

Property Value

Type: AdsVersion [▶ 642]

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.29.3 DeviceInfo Methods

The DeviceInfo [▶ 698] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

DeviceInfo Class [▶ 698]

TwinCAT.Ads Namespace [▶ 151]

6.2.30 IAdsAnyAccess Interface

Interface for accessing ADS 'Any' objects.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsAnyAccess
```

Methods

	Name	Description
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously.
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device.

	Name	Description
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1 IAdsAnyAccess Methods

Methods

	Name	Description
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object.

	Name	Description
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address.
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously.
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device.
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

	Name	Description
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.1 IAdsAnyAccess.ReadAny Method

Overload List

	Name	Description
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny.T. Method (UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadAny<T>(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny.T. Method (UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadAny<T>(
    uint variableHandle,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the value to read.

Return Value

Type: T
The value of the read symbol.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny Method (UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAny(
    uint variableHandle,
    Type type
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read object.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny.T. Method (UInt32, UInt32)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadAny<T>(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny Method (UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAny(
    uint variableHandle,
    Type type,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny.T. Method (UInt32, UInt32, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadAny<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.

args	Type: .System.Int32. Additional arguments.

Type Parameters

T	The type of the object to be read.

Return Value

Type: T
The read value.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny Method (UInt32, UInt32, Type)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.

Return Value

Type: Object
The read value.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAny Method (UInt32, UInt32, Type, .Int32.)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.

Return Value

Type: Object
The read value.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAny Overload [▶ 707]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.2 IAdsAnyAccess.ReadAnyAsync Method

Overload List

	Name	Description
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device.
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object.
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object.
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync.T. Method (UInt32, CancellationToken)

Reads data synchronously from an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable/symbol handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The Type of the value to be read.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync.T. Method (UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint variableHandle,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	Type of the object to be read

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, Type, CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable/symbol handle.
type	Type: System.Type Type of the object to be read.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync.T. Method (UInt32, UInt32, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T

Return Value

Type: Task.ResultValue [▶ 1029].T..
The asynchronous result.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, Type, .Int32., CancellationToken)

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyAsync(
    uint variableHandle,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync.T. Method (UInt32, UInt32, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadAnyAsync<T>(
    uint indexGroup,
    uint indexOffset,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The type of the result value.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, UInt32, Type, CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyAsync Method (UInt32, UInt32, Type, .Int32., CancellationToken)

Reads data asynchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyAsync Overload [▶ 714]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.3 IAdsAnyAccess.ReadAnyString Method

Overload List

	Name	Description
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable.
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyString Method (UInt32, Int32, Encoding)

Reads a string from the specified symbol/variable.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ReadAnyString(
    uint variableHandle,
    int len,
    Encoding encoding
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
The string value.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyString Overload [▶ 722]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyString Method (UInt32, UInt32, Int32, Encoding)

Reads as string from a specified address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ReadAnyString(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: String
System.String.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyString Overload [▶ 722]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.4 IAdsAnyAccess.ReadAnyStringAsync Method

Overload List

	Name	Description
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyStringAsync Method (UInt32, Int32, Encoding, CancellationToken)

Reads a string asynchronously from the specified symbol/variable

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyStringAsync(
    uint variableHandle,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 The variable handle.
len	Type: System.Int32 The length.

encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read string (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyStringAsync Overload [▶ 724]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.ReadAnyStringAsync Method (UInt32, UInt32, Int32, Encoding, CancellationToken)

Reads a string from a specified address asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadAnyStringAsync(
    uint indexGroup,
    uint indexOffset,
    int len,
    Encoding encoding,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.
len	Type: System.Int32 The string length to be read.
encoding	Type: System.Text.Encoding The encoding.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsAnyAccess Interface [▶ 702]

ReadAnyStringAsync Overload [▶ 724]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.5 IAdsAnyAccess.WriteAny Method

Overload List

	Name	Description
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device.
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAny Method (UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAny(
    uint variableHandle,
    Object value
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAny Overload [▶ 726]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAny Method (UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAny(
    uint variableHandle,
    Object value,
    int[] args
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAny Overload [▶ 726]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAny Method (UInt32, UInt32, Object)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAny Overload [▶ 726]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAny Method (UInt32, UInt32, Object, .Int32.)

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.

value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.

Remarks

If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAny Overload [▶ 726]

TwinCAT.Ads Namespace [▶ 151]

6.2.30.1.6 IAdsAnyAccess.WriteAnyAsync Method

Overload List

	Name	Description
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

	Name	Description
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Reference

IAdsAnyAccess Interface [▶ 702]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAnyAsync Method (UInt32, Object, CancellationToken)

Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAnyAsync Overload [▶ 729]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAnyAsync Method (UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAnyAsync(
    uint variableHandle,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAnyAsync Overload [▶ 729]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAnyAsync Method (UInt32, UInt32, Object, CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAnyAsync Overload [▶ 729]

TwinCAT.Ads Namespace [▶ 151]

IAdsAnyAccess.WriteAnyAsync Method (UInt32, UInt32, Object, .Int32., CancellationToken)

Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAnyAsync(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device.
args	Type: .System.Int32. Additional arguments.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous task operation. The result parameter ResultWrite [▶ 1032] of the write operation contains the ErrorCode [▶ 992].

Remarks

As object types only primitive types are supported. If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args.

Type of value Parameter	Necessary Arguments (args)
string	args[0]: Number of characters in the string typed as FixedLengthZeroTerminated [▶ 2395].
string[]	
Array	

Reference

IAdsAnyAccess Interface [▶ 702]

WriteAnyAsync Overload [▶ 729]

TwinCAT.Ads Namespace [▶ 151]

6.2.31 IAdsConnectAddress Interface

Interface for method to connect the ADS client via AMS Address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsConnectAddress : IAdsConnection,
    IConnection, IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess, IAdsAnyAccess,
    IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider, IAdsStateControl,
    IAdsSymbolChangedProvider, IAdsRpcInvoke
```

The IAdsConnectAddress type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server. (Inherited from IAdsConnection [▶ 765].)
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client. (Inherited from IAdsConnection [▶ 765].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection [▶ 765].)
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Connect. [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	Connect(Int32) [▶ 763]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 763]	Connects the target ADS Device.
	Connect(String, Int32) [▶ 764]	Connects to the target ADS Device.
	Connect(AmsNetId, Int32) [▶ 764]	Connects to the target ADS Device.
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier.., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)

	Name	Description
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	WriteValue.T. (ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T. (String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T. (ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T. (String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T. (String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T. (String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T. (String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(ISchedulr) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.31.1 IAdsConnectAddress Properties

The IAdsConnectAddress [▶ 733] type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server. (Inherited from IAdsConnection [▶ 765].)
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client. (Inherited from IAdsConnection [▶ 765].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)

	Name	Description
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection [▶ 765].)
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Reference

IAdsConnectAddress Interface [▶ 733]

TwinCAT.Ads Namespace [▶ 151]

6.2.31.2 IAdsConnectAddress Methods

The IAdsConnectAddress [▶ 733] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Connect. [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	Connect(Int32) [▶ 763]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 763]	Connects the target ADS Device.

	Name	Description
	Connect(String, Int32) [▶ 764]	Connects to the target ADS Device.
	Connect(AmsNetId, Int32) [▶ 764]	Connects to the target ADS Device.
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)

	Name	Description
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)

	Name	Description
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)

	Name	Description
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

IAdsConnectAddress Interface [▶ 733]

TwinCAT.Ads Namespace [▶ 151]

6.2.31.2.1 IAdsConnectAddress.Connect Method

Overload List

	Name	Description
	Connect. [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	Connect(Int32) [▶ 763]	Connects to the local target ADS Device.
	Connect(AmsAddress) [▶ 763]	Connects the target ADS Device.
	Connect(String, Int32) [▶ 764]	Connects to the target ADS Device.

	Name	Description
	Connect(AmsNetId, Int32) [▶ 764]	Connects to the target ADS Device.

Reference

IAdsConnectAddress Interface [▶ 733]

TwinCAT.Ads Namespace [▶ 151]

IAdsConnectAddress.Connect Method (Int32)

Connects to the local target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Connect(
    int port
)
```

Parameters

port	Type: System.Int32 The port number of the local ADS target device to connect to.

Reference

IAdsConnectAddress Interface [▶ 733]

Connect Overload [▶ 762]

TwinCAT.Ads Namespace [▶ 151]

IAdsConnectAddress.Connect Method (AmsAddress)

Connects the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Connect(
    AmsAddress address
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address of the target device.

Reference

IAdsConnectAddress Interface [▶ 733]

Connect Overload [▶ 762]

TwinCAT.Ads Namespace [▶ 151]

IAdsConnectAddress.Connect Method (String, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Connect(
    string netId,
    int port
)
```

Parameters

netId	Type: System.String The AmsNetId [▶ 665] of the ADS target device specified as string.
port	Type: System.Int32 The port number of the ADS target device.

Reference

IAdsConnectAddress Interface [▶ 733]

Connect Overload [▶ 762]

TwinCAT.Ads Namespace [▶ 151]

IAdsConnectAddress.Connect Method (AmsNetId, Int32)

Connects to the target ADS Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Connect(
    AmsNetId netId,
    int port
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The AmsNetId of the target device.

port	Type: System.Int32 The Ams Port number on the target device to connect to.

Reference

IAdsConnectAddress Interface [▶ 733]

Connect Overload [▶ 762]

TwinCAT.Ads Namespace [▶ 151]

6.2.31.3 IAdsConnectAddress Events

The IAdsConnectAddress [▶ 733] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

IAdsConnectAddress Interface [▶ 733]

TwinCAT.Ads Namespace [▶ 151]

6.2.32 IAdsConnection Interface

ADS Connection interface

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsConnection : IConnection,
    IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess, IAdsAnyAccess, IAdsHandle,
    IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider, IAdsStateControl, IAdsSymbolChangedProvider,
    IAdsRpcInvoke
```

The IAdsConnection type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server.
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client.
 	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Connect [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server.
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)

	Name	Description
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)

	Name	Description
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)

	Name	Description
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	WriteValues.T. (String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Remarks

The ADS connection interface represents the the logical point-to-point exception between Client and Server within an ADS Session.

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.IConnection [▶ 74]

TwinCAT.Ads.IAdsNotifications [▶ 839]

TwinCAT.Ads.IAdsAnyAccess [▶ 702]

TwinCAT.Ads.IAdsHandle [▶ 827]

TwinCAT.Ads.IAdsReadWrite2 [▶ 875]

TwinCAT.Ads.IAdsStateControl [▶ 915]

6.2.32.1 IAdsConnection Properties

The IAdsConnection [▶ 765] type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server.
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client.
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)

	Name	Description
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.1.1 IAdsConnection.Address Property

Gets the AmsAddress [▶ 648] of the ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.1.2 IAdsConnection.ClientAddress Property

Get the AmsAddress [▶ 648] of the ADS client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsAddress ClientAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.1.3 IAdsConnection.IsLocal Property

Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsLocal { get; }
```

Property Value

Type: Boolean

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.2 IAdsConnection Methods

The IAdsConnection [▶ 765] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)

	Name	Description
	Connect [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server.
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server.
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)

	Name	Description
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)

	Name	Description
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)

	Name	Description
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.2.1 IAdsConnection.ReadDeviceInfo Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DeviceInfo ReadDeviceInfo()
```

Return Value

Type: DeviceInfo [▶ 698]
DeviceInfo struct containing the name of the device and the version information.

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.2.2 IAdsConnection.ReadDeviceInfoAsync Method

Reads the identification and version number of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultDeviceInfo> ReadDeviceInfoAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultDeviceInfo [▶ 1001].
A task that represents the asynchronous 'ReadDeviceState' operation. The ResultDeviceInfo [▶ 1001] parameter contains the value DeviceInfo [▶ 1003] and the ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.32.3 IAdsConnection Events

The IAdsConnection [▶ 765] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

IAdsConnection Interface [▶ 765]

TwinCAT.Ads Namespace [▶ 151]

6.2.33 IAdsDisposableConnection Interface

Interface IAdsDisposableConnection Implements the IAdsConnectAddress [▶ 733] Implements the IRouterNotificationProvider [▶ 973] Implements the IAdsSymbolChangedProvider [▶ 934] Implements the IDisposable

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsDisposableConnection : IAdsConnectAddress,
    IAdsConnection, IConnection, IConnectionStateProvider, IAdsNotifications, IAdsSymbolicAccess,
    IAdsAnyAccess, IAdsHandle, IAdsReadWrite2, IAdsReadWrite, IAdsStateProvider,
    IAdsStateControl, IAdsSymbolChangedProvider, IAdsRpcInvoke, IRouterNotificationProvider, IDispos
able
```

The IAdsDisposableConnection type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server. (Inherited from IAdsConnection [▶ 765].)
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client. (Inherited from IAdsConnection [▶ 765].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)

	Name	Description
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsDisposed [▶ 812]	Gets a value indicating whether this instance is disposed.
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection [▶ 765].)
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Connect. [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	Connect(Int32) [▶ 763]	Connects to the local target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	Connect(AmsAddress) [▶ 763]	Connects the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	Connect(String, Int32) [▶ 764]	Connects to the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	Connect(AmsNetId, Int32) [▶ 764]	Connects to the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)

	Name	Description
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from IDisposable.)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)

	Name	Description
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)

	Name	Description
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)

	Name	Description
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)
	RouterStateChanged [▶ 973]	Router state changed event. (Inherited from IRouterNotificationProvider [▶ 973].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(ISchedulr) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.IAdsConnectAddress [▶ 733]

System.IDisposable

6.2.33.1 IAdsDisposableConnection Properties

The IAdsDisposableConnection [▶ 797] type exposes the following members.

Properties

	Name	Description
	Address [▶ 781]	Gets the AmsAddress [▶ 648] of the ADS server. (Inherited from IAdsConnection [▶ 765].)
	ClientAddress [▶ 781]	Get the AmsAddress [▶ 648] of the ADS client. (Inherited from IAdsConnection [▶ 765].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	DefaultValueEncoding [▶ 76]	Gets the default value encoding. (Inherited from IConnection [▶ 74].)
	Id [▶ 76]	Gets the Connection Identifier . (Inherited from IConnection [▶ 74].)
	IsConnected [▶ 76]	Gets a value indicating whether the local ADS port was opened successfully. It does not indicate if the target port is available. Use the method ReadState to determine if the target port is available. (Inherited from IConnection [▶ 74].)
	IsDisposed [▶ 812]	Gets a value indicating whether this instance is disposed.
	IsLocal [▶ 782]	Gets a value indicating whether the ADS client is connected to a ADS Server on the local computer. (Inherited from IAdsConnection [▶ 765].)
	Session [▶ 77]	Gets the session that initiated this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Timeout [▶ 77]	Gets the timeout (in milliseconds) (Inherited from IConnection [▶ 74].)

Reference

IAdsDisposableConnection Interface [▶ 797]

TwinCAT.Ads Namespace [▶ 151]

6.2.33.1.1 IAdsDisposableConnection.IsDisposed Property

Gets a value indicating whether this instance is disposed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsDisposed { get; }
```

Property Value

Type: Boolean
true if this instance is disposed; otherwise, false.

Reference

IAdsDisposableConnection Interface [▶ 797]

TwinCAT.Ads Namespace [▶ 151]

6.2.33.2 IAdsDisposableConnection Methods

The IAdsDisposableConnection [▶ 797] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache. (Inherited from IAdsSymbolicAccess [▶ 937].)
	Close [▶ 78]	Closes this IConnection [▶ 74] (Inherited from IConnection [▶ 74].)
	Connect. [▶ 78]	(Re)Connects the IConnection [▶ 74] when disconnected. (Inherited from IConnection [▶ 74].)
	Connect(Int32) [▶ 763]	Connects to the local target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	Connect(AmsAddress) [▶ 763]	Connects the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)

	Name	Description
	Connect(String, Int32) [▶ 764]	Connects to the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	Connect(AmsNetId, Int32) [▶ 764]	Connects to the target ADS Device. (Inherited from IAdsConnectAddress [▶ 733].)
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously. (Inherited from IAdsNotifications [▶ 839].)
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously. (Inherited from IAdsHandle [▶ 827].)
	Disconnect [▶ 79]	Disconnects this IConnection [▶ 74]. (Inherited from IConnection [▶ 74].)
	Dispose	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from IDisposable.)
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously (Inherited from IAdsRpcInvoke [▶ 886].)
	Read(UInt32, Memory) [▶ 832]	(Inherited from IAdsHandle [▶ 827].)
	Read(UInt32, UInt32, Memory) [▶ 876]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadAny(UInt32, Type) [▶ 709]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, Type, .Int32.) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type) [▶ 712]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny(UInt32, UInt32, Type, .Int32.) [▶ 713]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32) [▶ 707]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, .Int32.) [▶ 708]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32) [▶ 710]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAny.T.(UInt32, UInt32, .Int32.) [▶ 711]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, Type, CancellationToken) [▶ 716]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyAsync(UInt32, Type, .Int32., CancellationToken) [▶ 718]	Reads data synchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, CancellationToken) [▶ 720]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync(UInt32, UInt32, Type, .Int32., CancellationToken) [▶ 721]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, CancellationToken) [▶ 715]	Reads data synchronously from an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, .Int32., CancellationToken) [▶ 715]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, CancellationToken) [▶ 717]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyAsync.T.(UInt32, UInt32, .Int32., CancellationToken) [▶ 719]	Reads data asynchronously from an ADS device and writes it to an object. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, Int32, Encoding) [▶ 722]	Reads a string from the specified symbol/variable. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyString(UInt32, UInt32, Int32, Encoding) [▶ 723]	Reads as string from a specified address. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAnyStringAsync(UInt32, Int32, Encoding, CancellationToken) [▶ 724]	Reads a string asynchronously from the specified symbol/variable (Inherited from IAdsAnyAccess [▶ 702].)

	Name	Description
	ReadAnyStringAsync(UInt32, UInt32, Int32, Encoding, CancellationToken) [▶ 725]	Reads a string from a specified address asynchronously. (Inherited from IAdsAnyAccess [▶ 702].)
	ReadAsync(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadAsync(UInt32, UInt32, Memory, Void) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadDeviceInfo [▶ 795]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadDeviceInfoAsync [▶ 796]	Reads the identification and version number of an ADS server. (Inherited from IAdsConnection [▶ 765].)
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server. (Inherited from IAdsStateProvider [▶ 929].)
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	ReadWrite(UInt32, Memory, Void) [▶ 833]	(Inherited from IAdsHandle [▶ 827].)
	ReadWrite(UInt32, UInt32, Memory, Void) [▶ 877]	(Inherited from IAdsReadWrite2 [▶ 875].)
	ReadWriteAsync(UInt32, Memory, Void, Byte) [▶ 834]	(Inherited from IAdsHandle [▶ 827].)
	ReadWriteAsync(UInt32, UInt32, Memory, Void, Byte) [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event. (Inherited from IAdsNotifications [▶ 839].)
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously. (Inherited from IAdsHandle [▶ 827].)

	Name	Description
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification. (Inherited from IAdsNotifications [▶ 839].)
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously. (Inherited from IAdsHandle [▶ 827].)
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method. (Inherited from IAdsRpcInvoke [▶ 886].)
	TryRead(UInt32, Memory, Void) [▶ 836]	(Inherited from IAdsHandle [▶ 827].)
	TryRead(UInt32, UInt32, Memory, Void) [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful. (Inherited from IAdsStateProvider [▶ 929].)
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryReadWrite(UInt32, Memory, Void, Byte) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryReadWrite(UInt32, UInt32, Memory, Void, Byte) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite(UInt32, ReadOnlyMemory) [▶ 837]	(Inherited from IAdsHandle [▶ 827].)
	TryWrite(UInt32, UInt32, ReadOnlyMemory) [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	(Inherited from IAdsStateControl [▶ 915].)
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation. (Inherited from IAdsStateProvider [▶ 929].)

	Name	Description
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	Write(UInt32, ReadOnlyMemory) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address. (Inherited from IAdsReadWrite2 [▶ 875].)
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	(Inherited from IAdsReadWrite2 [▶ 875].)
	WriteAny(UInt32, Object) [▶ 726]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, Object, .Int32.) [▶ 727]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAny(UInt32, UInt32, Object, .Int32.) [▶ 728]	Writes an object synchronously to an ADS device. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, CancellationToken) [▶ 730]	Writes an object synchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, Object, .Int32., CancellationToken) [▶ 731]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAnyAsync(UInt32, UInt32, Object, .Int32., CancellationToken) [▶ 732]	Writes an object asynchronously to an ADS device. If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. (Inherited from IAdsAnyAccess [▶ 702].)
	WriteAsync(UInt32, ReadOnlyMemory, Void) [▶ 838]	(Inherited from IAdsHandle [▶ 827].)

	Name	Description
	WriteAsync(UInt32, UInt32, ReadOnlyMemory, Void) [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server. (Inherited from IAdsStateControl [▶ 915].)
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	(Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously. (Inherited from IAdsStateControl [▶ 915].)
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	(Inherited from IAdsStateControl [▶ 915].)
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol. (Inherited from IAdsSymbolicAccess [▶ 937].)

Extension Methods

	Name	Description
	PollAdsState(IObservable.Unit.) [▶ 1061]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsState(TimeSpan) [▶ 1062]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(IObservable.Unit., CancellationToken) [▶ 1064]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollAdsStateAsync(TimeSpan, CancellationToken) [▶ 1065]	Overloaded. Gets an observable sequence of AdsState [▶ 626]s via Polling. (Defined by AdsClientExtensions [▶ 1056].)
	PollValues(String, Type, IObservable.Unit.) [▶ 1089]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan) [▶ 1090]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan) [▶ 1093]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues(String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit.) [▶ 1083]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)

	Name	Description
	PollValues.T.(String, TimeSpan) [▶ 1084]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, TimeSpan, Func.Exception, T.) [▶ 1088]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit.) [▶ 1085]	Overloaded. Polls the symbol values on time points where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan) [▶ 1086]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Overloaded. Polls the symbol values on timepoints where the polling observable streams data / triggers (Defined by AnyTypeExtensions [▶ 1075].)
	PollValues.T.(String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Overloaded. Polls the symbol as value sequence of object values with a specified period time. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol) [▶ 1068]	Overloaded. Gets an observable sequence of Notification [▶ 974]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection) [▶ 1069]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbol, NotificationSettings) [▶ 1070]	Overloaded. Gets an observable sequence of SymbolValueNotification [▶ 1104]s. (Defined by AdsClientExtensions [▶ 1056].)
	WhenNotification(ISymbolCollection, NotificationSettings) [▶ 1071]	Overloaded. Gets an observable sequence of Notification [▶ 974] objects. (Defined by AdsClientExtensions [▶ 1056].)

	Name	Description
	WhenNotification(String, Type, NotificationSettings) [▶ 1099]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenNotification.T.(String, NotificationSettings) [▶ 1098]	Overloaded. Creates an observable sequence of values that are created by ADS Notifications. (Defined by AnyTypeExtensions [▶ 1075].)
	WhenSymbolVersionChanges. [▶ 1073]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenSymbolVersionChanges(IScheduler) [▶ 1074]	Overloaded. Gets an observable sequence of SymbolVersion changed counts. (Defined by AdsClientExtensions [▶ 1056].)
	WhenValueChanged [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues.T.(String, IObservable.T.) [▶ 1101]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)
	WriteValues.T.(String, IObservable.T., Action.Exception.) [▶ 1102]	Overloaded. Writes the sequence of values to the symbol specified by the instance path. (Defined by AnyTypeExtensions [▶ 1075].)

Reference

IAdsDisposableConnection Interface [▶ 797]

TwinCAT.Ads Namespace [▶ 151]

6.2.33.3 IAdsDisposableConnection Events

The IAdsDisposableConnection [▶ 797] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management. (Inherited from IAdsNotifications [▶ 839].)
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client. (Inherited from IAdsNotifications [▶ 839].)

	Name	Description
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed. (Inherited from IAdsStateProvider [▶ 929].)
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications) (Inherited from IAdsNotifications [▶ 839].)
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed. (Inherited from IAdsSymbolChangedProvider [▶ 934].)
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)
	RouterStateChanged [▶ 973]	Router state changed event. (Inherited from IRouterNotificationProvider [▶ 973].)

Reference

IAdsDisposableConnection Interface [▶ 797]

TwinCAT.Ads Namespace [▶ 151]

6.2.34 IAdsHandle Interface

Interface for ADS access via variable handle

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsHandle
```

The IAdsHandle type exposes the following members.

Methods

	Name	Description
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously.
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously.
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously.
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously.
	Read [▶ 832]	

	Name	Description
	ReadAsync [▶ 833]	
	ReadWrite [▶ 833]	
	ReadWriteAsync [▶ 834]	
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously.
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously.
	TryRead [▶ 836]	
	TryReadWrite [▶ 837]	
	TryWrite [▶ 837]	
	Write [▶ 838]	
	WriteAsync [▶ 838]	

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1 IAdsHandle Methods

The IAdsHandle [▶ 827] type exposes the following members.

Methods

	Name	Description
	CreateVariableHandle [▶ 829]	Determines the Symbol handle by its instance path synchronously.
	CreateVariableHandleAsync [▶ 830]	Determines the Symbol handle by its instance path asynchronously.
	DeleteVariableHandle [▶ 830]	Releases the specified symbol/variable handle synchronously.
	DeleteVariableHandleAsync [▶ 831]	Releases the specified symbol/variable handle asynchronously.
	Read [▶ 832]	
	ReadAsync [▶ 833]	
	ReadWrite [▶ 833]	
	ReadWriteAsync [▶ 834]	

	Name	Description
	TryCreateVariableHandle [▶ 834]	Determines the Symbol handle by its instance path synchronously.
	TryDeleteVariableHandle [▶ 835]	Releases the specified symbol/variable handle synchronously.
	TryRead [▶ 836]	
	TryReadWrite [▶ 837]	
	TryWrite [▶ 837]	
	Write [▶ 838]	
	WriteAsync [▶ 838]	

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.1 IAdsHandle.CreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint CreateVariableHandle(
    string symbolPath
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.

Return Value

Type: UInt32
The symbols/variable handle

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandle(String) is the DeleteVariableHandle(UInt32) [▶ 830]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.DeleteVariableHandle(UInt32) [▶ 830]

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [▶ 834]

6.2.34.1.2 IAdsHandle.CreateVariableHandleAsync Method

Determines the Symbol handle by its instance path asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultHandle> CreateVariableHandleAsync(
    string symbolPath,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'CreateVariableHandle' operation. The ResultHandle [▶ 1005] parameter contains the variable handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this CreateVariableHandleAsync(String, CancellationToken) is the DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [▶ 834]

IAdsHandle.CreateVariableHandle(String) [▶ 829]

6.2.34.1.3 IAdsHandle.DeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void DeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Return Value

Type:
The ADS error code.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) [▶ 835] is the TryCreateVariableHandle(String, UInt32.) [▶ 834]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.CreateVariableHandle(String) [▶ 829]

IAdsHandle.TryDeleteVariableHandle(UInt32) [▶ 835]

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

6.2.34.1.4 IAdsHandle.DeleteVariableHandleAsync Method

Releases the specified symbol/variable handle asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> DeleteVariableHandleAsync(
    uint variableHandle,
    CancellationToken cancel
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'DeleteVariableHandle' operation. The ResultAds [▶ 989] parameter contains the ErrorCode [▶ 992] after execution.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this DeleteVariableHandleAsync(UInt32, CancellationToken) is the CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

IAdsHandle.TryDeleteVariableHandle(UInt32) [▶ 835]

IAdsHandle.DeleteVariableHandle(UInt32) [▶ 830]

6.2.34.1.5 IAdsHandle.Read Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Read(
    uint variableHandle,
    Memory readBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory

Return Value

Type: Int32

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.6 IAdsHandle.ReadAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRead> ReadAsync(
    uint variableHandle,
    Memory readBuffer,
    void cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.7 IAdsHandle.ReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.8 IAdsHandle.ReadWriteAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadWrite> ReadWriteAsync(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.9 IAdsHandle.TryCreateVariableHandle Method

Determines the Symbol handle by its instance path synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryCreateVariableHandle(
    string symbolPath,
    out uint variableHandle
)
```

Parameters

symbolPath	Type: System.String SymbolName / InstancePath.
variableHandle	Type: System.UInt32. The symbols handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryCreateVariableHandle(String, UInt32.) is the TryDeleteVariableHandle(UInt32) [▶ 835]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.TryDeleteVariableHandle(UInt32) [▶ 835]

IAdsHandle.CreateVariableHandleAsync(String, CancellationToken) [▶ 830]

IAdsHandle.CreateVariableHandle(String) [▶ 829]

6.2.34.1.10 IAdsHandle.TryDeleteVariableHandle Method

Releases the specified symbol/variable handle synchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryDeleteVariableHandle(
    uint variableHandle
)
```

Parameters

variableHandle	Type: System.UInt32 Handle of the ADS variable

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Remarks

It is a good practice to release all variable handles after use to regain internal resources in the TwinCAT subsystem. The composite method to this TryDeleteVariableHandle(UInt32) is the TryCreateVariableHandle(String, UInt32.) [▶ 834]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

IAdsHandle.TryCreateVariableHandle(String, UInt32.) [▶ 834]

IAdsHandle.DeleteVariableHandleAsync(UInt32, CancellationToken) [▶ 831]

IAdsHandle.DeleteVariableHandle(UInt32) [▶ 830]

6.2.34.1.11 IAdsHandle.TryRead Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryRead(
    uint variableHandle,
    Memory readBuffer,
    void readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.12 IAdsHandle.TryReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadWrite(
    uint variableHandle,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

variableHandle	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.13 IAdsHandle.TryWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWrite(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.14 IAdsHandle.Write Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Write(
    uint variableHandle,
    ReadOnlyMemory writeBuffer
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.34.1.15 IAdsHandle.WriteAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAsync(
    uint variableHandle,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

variableHandle	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

IAdsHandle Interface [▶ 827]

TwinCAT.Ads Namespace [▶ 151]

6.2.35 IAdsNotifications Interface

Interface for Notification management.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsNotifications
```

The IAdsNotifications type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.

	Name	Description
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification.

	Name	Description
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously.
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification.

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client.
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.35.1 IAdsNotifications Methods

The IAdsNotifications [▶ 839] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

	Name	Description
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	DeleteDeviceNotification [▶ 857]	Deletes a registered notification.
	DeleteDeviceNotificationAsync [▶ 858]	Deletes a registered notification asynchronously.
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

	Name	Description
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryDeleteDeviceNotification [▶ 864]	Deletes a registered notification.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

6.2.35.1.1 IAdsNotifications.AddDeviceNotification Method

Overload List

	Name	Description
	AddDeviceNotification(String, Int32, NotificationSettings, Object) [▶ 844]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object) [▶ 845]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AddDeviceNotification Method (String, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

symbolPath	Type: System.String Symbol / Instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotification Overload [▶ 844]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

IAdsNotifications.AddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

Return Value

Type: UInt32
The notification handle.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotification Overload [▶ 844]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.35.1.2 IAdsNotifications.AddDeviceNotificationAsync Method

Overload List

	Name	Description
	AddDeviceNotificationAsync(String, Int32, NotificationSettings, Object, CancellationToken) [▶ 847]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.
	AddDeviceNotificationAsync(UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken) [▶ 848]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AddDeviceNotificationAsync Method (String, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultHandle> AddDeviceNotificationAsync(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.

dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always be called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationAsync Overload [▶ 847]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

IAdsNotifications.AddDeviceNotificationAsync Method (UInt32, UInt32, Int32, NotificationSettings, Object, CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultHandle> AddDeviceNotificationAsync(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always be called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationAsync Overload [▶ 847]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

6.2.35.1.3 IAdsNotifications.AddDeviceNotificationEx Method

Overload List

	Name	Description
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type) [▶ 850]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32.) [▶ 851]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type) [▶ 852]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32.) [▶ 853]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationEx Overload [▶ 850]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

IAdsNotifications.AddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')

Return Value

Type: UInt32
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationEx Overload [▶ 850]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

IAdsNotifications.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')

Return Value

Type: UInt32
The notification handle.

Remarks

xxxxxBecause notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.
AddDeviceNotificationEx Overload [▶ 850]

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationEx Overload [▶ 850]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

IAdsNotifications.AddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification event. If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive ('AnyType') types are allowed for the parameter type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint AddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument.
args	Type: .System.Int32. Additional arguments for 'AnyType' types.

Return Value

Type: UInt32
The notification handle.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationEx Overload [▶ 850]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotificationEx [▶ 868]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

6.2.35.1.4 IAdsNotifications.AddDeviceNotificationExAsync Method

Overload List

	Name	Description
	AddDeviceNotificationExAsync(String, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 855]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	AddDeviceNotificationExAsync(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken) [▶ 856]	Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AddDeviceNotificationExAsync Method (String, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultHandle> AddDeviceNotificationExAsync(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)

type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always be called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationExAsync Overload [▶ 855]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

IAdsNotifications.AddDeviceNotificationExAsync Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., CancellationToken)

Connects a variable to the ADS client asynchronously. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultHandle> AddDeviceNotificationExAsync(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    CancellationToken cancel
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.
userData	Type: System.Object This object can be used to store user specific data.
type	Type: System.Type Type of the object stored in the event argument, only Primitive 'AnyTypes' allowed.
args	Type: .System.Int32. Additional arguments (for 'AnyType')
cancel	Type: System.Threading.CancellationToken The Cancellation token.

Return Value

Type: Task.ResultHandle [▶ 1005].
A task that represents the asynchronous 'AddDeviceNotification' operation. The ResultHandle [▶ 1005] type parameter contains the created handle (Handle [▶ 1007]) and the ErrorCode [▶ 992] after execution.

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

AddDeviceNotificationExAsync Overload [▶ 855]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.DeleteDeviceNotificationAsync(UInt32, CancellationToken) [▶ 858]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.35.1.5 IAdsNotifications.DeleteDeviceNotification Method

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void DeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.

Remarks

This is the complementary method to AddDeviceNotification Overload [▶ 844] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.35.1.6 IAdsNotifications.DeleteDeviceNotificationAsync Method

Deletes a registered notification asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> DeleteDeviceNotificationAsync(
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'DeleteDeviceNotification' operation. The ErrorCode [▶ 992] property contains the ADS error code after execution.

Remarks

This is the complementary method to AddDeviceNotificationAsync Overload [▶ 847] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotificationAsync Overload [▶ 847]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotification Overload [▶ 844]

6.2.35.1.7 IAdsNotifications.TryAddDeviceNotification Method

Overload List

	Name	Description
	TryAddDeviceNotification(String, Int32, NotificationSettings, Object, UInt32.) [▶ 859]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.
	TryAddDeviceNotification(UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.) [▶ 861]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.TryAddDeviceNotification Method (String, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryAddDeviceNotification(
    string symbolPath,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

symbolPath	Type: System.String The symbol/instance path of the ADS variable.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
userData	Type: System.Object This object can be used to store user specific data.
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS ErrorCode.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 864] should always be called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

TryAddDeviceNotification Overload [▶ 859]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [▶ 864]

AddDeviceNotification Overload [▶ 844]

AddDeviceNotificationAsync Overload [▶ 847]

TryAddDeviceNotification Overload [▶ 859]

IAdsNotifications.TryAddDeviceNotification Method (UInt32, UInt32, Int32, NotificationSettings, Object, UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotification [▶ 866] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryAddDeviceNotification(
    uint indexGroup,
    uint indexOffset,
    int dataSize,
    NotificationSettings settings,
    Object userData,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 The index group number of the requested ADS service.
indexOffset	Type: System.UInt32 The index offset number of the requested ADS service.
dataSize	Type: System.Int32 Maximum amount of data in bytes to receive with this ADS Notification.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Remarks

The

dataSize

Parameter defines the amount of bytes, that will be attached to the AdsNotification [▶ 866] as value. Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 864] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

TryAddDeviceNotification Overload [▶ 859]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.TryDeleteDeviceNotification(UInt32) [▶ 864]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotification Overload [▶ 844]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.35.1.8 IAdsNotifications.TryAddDeviceNotificationEx Method

Overload List

	Name	Description
	TryAddDeviceNotificationEx(String, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 862]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.
	TryAddDeviceNotificationEx(UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.) [▶ 863]	Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.TryAddDeviceNotificationEx Method (String, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryAddDeviceNotificationEx(
    string symbolPath,
    NotificationSettings settings,
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

Parameters

symbolPath	Type: System.String Symbol/Instance path of the ADS variable.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. Additional arguments (for 'AnyType')
handle	Type: System.UInt32. The notification handle

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Remarks

Because notifications allocate TwinCAT system resources, a complementary call to TryDeleteDeviceNotification(UInt32) [▶ 864] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

TryAddDeviceNotificationEx Overload [▶ 862]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

AddDeviceNotificationEx Overload [▶ 850]

AddDeviceNotificationExAsync Overload [▶ 855]

TryAddDeviceNotificationEx Overload [▶ 862]

IAdsNotifications.TryAddDeviceNotificationEx Method (UInt32, UInt32, NotificationSettings, Object, Type, .Int32., UInt32.)

Connects a variable to the ADS client. The ADS client will be notified by the AdsNotificationEx [▶ 868] event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryAddDeviceNotificationEx(
    uint indexGroup,
    uint indexOffset,
    NotificationSettings settings,
```

```
    Object userData,
    Type type,
    int[] args,
    out uint handle
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
userData	Type: System.Object This object can be used to store user specific data (tag data)
type	Type: System.Type Type of the object stored in the event argument ('AnyType')
args	Type: .System.Int32. The 'AnyType' arguments.
handle	Type: System.UInt32. The notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error code.

Remarks

If type is a string type, the first element of the parameter args specifies the number of characters of the string. If type is an array type, the number of elements for each dimension has to be specified in the parameter args. Only primitive types (AnyType) are supported by this method. Because notifications allocate TwinCAT system resources, a complementary call to DeleteDeviceNotification(UInt32) [▶ 857] should always called when the notification is not used anymore.

Reference

IAdsNotifications Interface [▶ 839]

TryAddDeviceNotificationEx Overload [▶ 862]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.DeleteDeviceNotification(UInt32) [▶ 857]

IAdsNotifications.AdsNotificationEx [▶ 868]

IAdsNotifications.AdsNotificationError [▶ 867]

AddDeviceNotificationEx Overload [▶ 850]

TryAddDeviceNotificationEx Overload [▶ 862]

AddDeviceNotificationExAsync Overload [▶ 855]

6.2.35.1.9 IAdsNotifications.TryDeleteDeviceNotification Method

Deletes a registered notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryDeleteDeviceNotification(
    uint notificationHandle
)
```

Parameters

notificationHandle	Type: System.UInt32 Notification handle.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code.

Remarks

This is the complementary method to TryAddDeviceNotification Overload [▶ 859] overloads and should be called when the notification is not needed anymore the free TwinCAT realtime resources.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

AddDeviceNotification Overload [▶ 844]

IAdsNotifications.AdsNotification [▶ 866]

TryAddDeviceNotification Overload [▶ 859]

AddDeviceNotificationAsync Overload [▶ 847]

6.2.35.2 IAdsNotifications Events

The IAdsNotifications [▶ 839] type exposes the following members.

Events

	Name	Description
	AdsNotification [▶ 866]	Occurs when the ADS device sends a notification to the client.
	AdsNotificationError [▶ 867]	Occurs when a exception has occurred during notification management.
	AdsNotificationEx [▶ 868]	Occurs when the ADS devices sends a notification to the client.

	Name	Description
	AdsSumNotification [▶ 868]	Occurs when Notifications are send (bundled notifications)

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

6.2.35.2.1 IAdsNotifications.AdsNotification Event

Occurs when the ADS device sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsNotificationEventArgs> AdsNotification
```

Value

Type: System.EventHandler.AdsNotificationEventArgs [▶ 597].

Remarks

The Event Argument contains the raw data value of the notification, not marshalled to .NET types.

Examples

Example of receiving AdsNotification events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsNotification += Client_AdsNotification2;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];
    int size = sizeof(UInt32);

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", size, new Notific
ationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
```

```
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_AdsNotification2(object sender, AdsNotificationEventArgs e)
{
    // Or here we know about UDINT type --> can be marshalled as UINT32
    uint nCounter = BinaryPrimitives.ReadUInt32LittleEndian(e.Data.Span);

    // If Synchronization is needed (e.g. in Windows.Forms or WPF applications)
    // we could synchronize via SynchronizationContext into the UI Thread

    /*SynchronizationContext syncContext = SynchronizationContext.Current;
      _context.Post(status => someLabel.Text = nCounter.ToString(), null); // Non-blocking post */
}
```

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

6.2.35.2.2 IAdsNotifications.AdsNotificationError Event

Occurs when a exception has occurred during notification management.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsNotificationErrorEventArgs> AdsNotificationError
```

Value

Type: System.EventHandler.AdsNotificationErrorEventArgs [▶ 595].

Remarks

The occurrence of this event can have two different reasons:

1. Indicates an internal error occurred during Notification management.
2. The registered notification becomes invalid on the server, eg. after a PLC Download / Online Change. If the ADS Server detects that the (still registered) Notification Sender is getting invalid, it sends an error notification so that the client will be informed about detached notifications. The event arguments contains the 'AdsInvalidNotificationException' which describes the invalid notification handle by its 'AdsInvalidNotificationException.Handle' property.

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

IAdsNotifications.AdsNotificationEx [▶ 868]

6.2.35.2.3 IAdsNotifications.AdsNotificationEx Event

Occurs when the ADS devices sends a notification to the client.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsNotificationExEventArgs> AdsNotificationEx
```

Value

Type: System.EventHandler.AdsNotificationExEventArgs [▶ 601].

Remarks

The Notification event arguments marshals the data value automatically to the specified .NET Type with ANY_TYPE marshallers.

Examples

Example of receiving AdsNotificationEx events.

Trigger on changed values by ADS Notifications

```
CancellationToken cancel = CancellationToken.None;

using (AdsClient client = new AdsClient())
{
    client.AdsNotificationEx += Client_AdsNotificationEx;
    client.Connect(AmsNetId.Local, 851);

    // Add UDINT
    ResultHandle resultHandle = await client.AddDeviceNotificationExAsync("MAIN.udint", new Notifica
tionSettings(AdsTransMode.OnChange, 200, 200), null, typeof(uint),null, cancel);
    await Task.Delay(5000, cancel); // Wait ....
    ResultAds resultHandleDelete = await client.DeleteDeviceNotificationAsync(resultHandle.Handle,ca
ncel); // Unregister Event
}
```

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

6.2.35.2.4 IAdsNotifications.AdsSumNotification Event

Occurs when Notifications are send (bundled notifications)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsSumNotificationEventArgs> AdsSumNotification
```

Value

Type: System.EventHandler.AdsSumNotificationEventArgs.

Remarks

As an optimization, this event receives all ADS Notifications that occurred at one point in time together. As consequence, the overhead of handler code is reduced, what can be important if notifications are triggered in a high frequency and the event has to be synchronized to the UI thread context. Because multiple notifications are bound together, less thread synchronization is necessary. The AdsNotification [▶ 866] and AdsNotificationEx [▶ 868] events shouldn't be used when SumNotifications are registered, because they have an performance side effect to this AdsSumNotification event. The full performance is reached only, when all notifications are handled on this event.

Examples

Example of receiving AdsSumNotification events.

Trigger on changed values by ADS Notifications

```
private async Task RegisterSumNotificationsAsync()
{
    CancellationToken cancel = CancellationToken.None;

    using (AdsClient client = new AdsClient())
    {
    // Add the Notification event handler
    client.AdsSumNotification += Client_SumNotification;

    // Connect to target
    client.Connect(AmsNetId.Local, 851);
    uint notificationHandle = 0;

    // Notification to a DINT Type (UINT32)
    // Check for change every 200 ms

    //byte[] notificationBuffer = new byte[sizeof(UInt32)];

    ResultHandle result = await client.AddDeviceNotificationAsync("MAIN.nCounter", sizeof(UInt32), n
ew NotificationSettings(AdsTransMode.OnChange, 200, 0), null, cancel);

    if (result.Succeeded)
    {
        notificationHandle = result.Handle;
        await Task.Delay(5000); // Wait asynchronously without blocking the UI Thread.
                    // Unregister the Event / Handle
        ResultAds result2 = await client.DeleteDeviceNotificationAsync(notificationHandle, cancel);
    }
    client.AdsNotification -= Client_AdsNotification2;
    }
}

private void Client_SumNotification(object sender, AdsSumNotificationEventArgs e)
{
    // Timestamp of the Notification List
    DateTimeOffset dateTime = e.TimeStamp;

    // List of Raw ADS Notifications
    IList<Notification> notifications = e.Notifications;

    foreach(Notification notification in notifications)
    {
    // Notifications can be handled more efficiently, because they occur togeterh
    // handler and can be transformed/
synchronized in one step compared to AdsClient.AdsNotifcation events.
    }
}
```

Reference

IAdsNotifications Interface [▶ 839]

TwinCAT.Ads Namespace [▶ 151]

IAdsNotifications.AdsNotification [▶ 866]

6.2.36 IAdsReadWrite Interface

Interface for ADS Read/Write access via IndexGroup / IndexOffset

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsReadWrite
```

The IAdsReadWrite type exposes the following members.

Methods

	Name	Description
	ReadAsync [▶ 871]	
	ReadWriteAsync [▶ 871]	
	TryRead [▶ 872]	
	TryReadWrite [▶ 873]	
	TryWrite [▶ 873]	
	WriteAsync [▶ 874]	

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1 IAdsReadWrite Methods

The IAdsReadWrite [▶ 870] type exposes the following members.

Methods

	Name	Description
	ReadAsync [▶ 871]	
	ReadWriteAsync [▶ 871]	
	TryRead [▶ 872]	
	TryReadWrite [▶ 873]	
	TryWrite [▶ 873]	
	WriteAsync [▶ 874]	

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.1 IAdsReadWrite.ReadAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRead> ReadAsync(
    uint indexGroup,
    uint indexOffset,
    Memory buffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
buffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultRead [▶ 1008].

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.2 IAdsReadWrite.ReadWriteAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadWrite> ReadWriteAsync(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
cancel	Type: System.Byte

Return Value

Type: Task.ResultReadWrite [▶ 1019].

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.3 IAdsReadWrite.TryRead Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory buffer,
    void readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
buffer	Type: Memory
readBytes	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.4 IAdsReadWrite.TryReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.5 IAdsReadWrite.TryWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.36.1.6 IAdsReadWrite.WriteAsync Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteAsync(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWrite [▶ 1032].

Reference

IAdsReadWrite Interface [▶ 870]

TwinCAT.Ads Namespace [▶ 151]

6.2.37 IAdsReadWrite2 Interface

Interface for ADS Read/Write access via IndexGroup / IndexOffset

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsReadWrite2 : IAdsReadWrite
```

The IAdsReadWrite2 type exposes the following members.

Methods

	Name	Description
	Read [▶ 876]	
	ReadAsync [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadWrite [▶ 877]	
	ReadWriteAsync [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	TryRead [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadWrite [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	
	WriteAsync [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.37.1 IAdsReadWrite2 Methods

The IAdsReadWrite2 [▶ 875] type exposes the following members.

Methods

	Name	Description
	Read [▶ 876]	
	ReadAsync [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	ReadWrite [▶ 877]	

	Name	Description
	ReadWriteAsync [▶ 871]	(Inherited from IAdsReadWrite [▶ 870].)
	TryRead [▶ 872]	(Inherited from IAdsReadWrite [▶ 870].)
	TryReadWrite [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	TryWrite [▶ 873]	(Inherited from IAdsReadWrite [▶ 870].)
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	
	WriteAsync [▶ 874]	(Inherited from IAdsReadWrite [▶ 870].)

Reference

IAdsReadWrite2 Interface [▶ 875]

TwinCAT.Ads Namespace [▶ 151]

6.2.37.1.1 IAdsReadWrite2.Read Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory

Return Value

Type: Int32

Reference

IAdsReadWrite2 Interface [▶ 875]

TwinCAT.Ads Namespace [▶ 151]

6.2.37.1.2 IAdsReadWrite2.ReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void

Return Value

Type: Int32

Reference

IAdsReadWrite2 Interface [▶ 875]

TwinCAT.Ads Namespace [▶ 151]

6.2.37.1.3 IAdsReadWrite2.Write Method

Overload List

	Name	Description
	Write(UInt32, UInt32) [▶ 878]	Triggers a 'Write' call to the ADS device at the specified address.
	Write(UInt32, UInt32, ReadOnlyMemory) [▶ 878]	

Reference

IAdsReadWrite2 Interface [▶ 875]

TwinCAT.Ads Namespace [▶ 151]

IAdsReadWrite2.Write Method (UInt32, UInt32)

Triggers a 'Write' call to the ADS device at the specified address.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Write(
    uint indexGroup,
    uint indexOffset
)
```

Parameters

indexGroup	Type: System.UInt32 The index group.
indexOffset	Type: System.UInt32 The index offset.

Reference

IAdsReadWrite2 Interface [▶ 875]

Write Overload [▶ 877]

TwinCAT.Ads Namespace [▶ 151]

IAdsReadWrite2.Write Method (UInt32, UInt32, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory

Reference

IAdsReadWrite2 Interface [▶ 875]

Write Overload [▶ 877]

TwinCAT.Ads Namespace [▶ 151]

6.2.38 IAdsReadWriteTimeoutAccess Interface

Interface IAdsReadWriteTimeoutAccess

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsReadWriteTimeoutAccess
```

The IAdsReadWriteTimeoutAccess type exposes the following members.

Methods

	Name	Description
	Read [▶ 880]	
	ReadAny [▶ 880]	Reads data synchronously from an ADS device and writes it to an object.
	ReadWrite [▶ 881]	
	TryRead [▶ 882]	
	TryReadWrite [▶ 883]	
	TryWrite [▶ 883]	
	Write [▶ 884]	
	WriteAny [▶ 885]	Writes an object synchronously to an ADS device.

Remarks

For internal use only.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1 IAdsReadWriteTimeoutAccess Methods

The IAdsReadWriteTimeoutAccess [▶ 879] type exposes the following members.

Methods

	Name	Description
	Read [▶ 880]	
	ReadAny [▶ 880]	Reads data synchronously from an ADS device and writes it to an object.
	ReadWrite [▶ 881]	

	Name	Description
	TryRead [▶ 882]	
	TryReadWrite [▶ 883]	
	TryWrite [▶ 883]	
	Write [▶ 884]	
	WriteAny [▶ 885]	Writes an object synchronously to an ADS device.

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.1 IAdsReadWriteTimeoutAccess.Read Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Read(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
timeout	Type: System.Void

Return Value

Type: Int32

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.2 IAdsReadWriteTimeoutAccess.ReadAny Method

Reads data synchronously from an ADS device and writes it to an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAny(
    uint indexGroup,
    uint indexOffset,
    Type type,
    int[] args,
    int timeout
)
```

Parameters

indexGroup	Type: System.UInt32 Index group of the ADS variable.
indexOffset	Type: System.UInt32 Index offset of the ADS variable.
type	Type: System.Type Type of the object to be read.
args	Type: .System.Int32. Additional arguments.
timeout	Type: System.Int32 The timeout.

Return Value

Type: Object
The Value of the data marshalled to the specified type.

Remarks

If the Type of the object to be read is a string type, the first element of the parameter args specifies the number of characters of the string. If the Type of the object to be read is an array type, the number of elements for each dimension has to be specified in the parameter args. The type is limited to Primitive types ('AnyType').

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.3 IAdsReadWriteTimeoutAccess.ReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
```

```
    void writeBuffer,
    byte timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
timeout	Type: System.Byte

Return Value

Type: Int32

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.4 IAdsReadWriteTimeoutAccess.TryRead Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryRead(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void timeout,
    byte readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
timeout	Type: System.Void
readBytes	Type: System.Byte

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.5 IAdsReadWriteTimeoutAccess.TryReadWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadWrite(
    uint indexGroup,
    uint indexOffset,
    Memory readBuffer,
    void writeBuffer,
    byte timeout,
    ReadOnlyMemory readBytes
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readBuffer	Type: Memory
writeBuffer	Type: System.Void
timeout	Type: System.Byte
readBytes	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.6 IAdsReadWriteTimeoutAccess.TryWrite Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWrite(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.7 IAdsReadWriteTimeoutAccess.Write Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Write(
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.38.1.8 IAdsReadWriteTimeoutAccess.WriteAny Method

Writes an object synchronously to an ADS device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAny(
    uint indexGroup,
    uint indexOffset,
    Object value,
    int[] args,
    int timeout
)
```

Parameters

indexGroup	Type: System.UInt32 Contains the index group number of the requested ADS service.
indexOffset	Type: System.UInt32 Contains the index offset number of the requested ADS service.
value	Type: System.Object Object to write to the ADS device (Primitive type, 'AnyType')
args	Type: .System.Int32. Additional arguments.
timeout	Type: System.Int32 The timeout.

Return Value

Type:
System.Int32.

Remarks

If the Type of the object to be written is a string type, the first element of parameter args specifies the number of characters of the string. If the Type of the object to be written is an array type, the number of elements for each dimension has to be specified in the parameter args. The value is limited to Primitive types ('AnyType').

Reference

IAdsReadWriteTimeoutAccess Interface [▶ 879]

TwinCAT.Ads Namespace [▶ 151]

6.2.39 IAdsRpcInvoke Interface

Interface IAdsRpcInvoke

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsRpcInvoke
```

Methods

	Name	Description
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method.

Remarks

This interface is used to invoke ADS RPC Method calls. This can be done 'symbolic' via the Method name or the MethodID of the method on the specified symbol. To activate the RPC Access within the PLC environment, its dataType (Structure, FB) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.39.1 IAdsRpcInvoke Methods

Methods

	Name	Description
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier.., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method.

Reference

IAdsRpcInvoke Interface [▶ 886]

TwinCAT.Ads Namespace [▶ 151]

6.2.39.1.1 IAdsRpcInvoke.InvokeRpcMethod Method

Overload List

	Name	Description
	InvokeRpcMethod(String, String, .Object.) [▶ 890]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .Object..) [▶ 891]	Invokes the specified RPC Method
	InvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 893]	Invokes the specified RPC Method

Reference

IAdsRpcInvoke Interface [▶ 886]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL

Return Value

Type: Object
The return value of the Method (as object).

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
```

```
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethod Overload [▶ 889]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL
outParameters	Type: .System.Object.. The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
```

```
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethod Overload [▶ 889]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifiying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifiying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethod Overload [▶ 889]

TwinCAT.Ads Namespace [▶ 151]

6.2.39.1.2 IAdsRpcInvoke.InvokeRpcMethodAsync Method

Overload List

	Name	Description
	InvokeRpcMethodAsync(String, String, .Object., CancellationToken) [▶ 895]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 897]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 899]	Invokes the specified RPC Method asynchronously

Reference

IAdsRpcInvoke Interface [▶ 886]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethodAsync Method (String, String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
```

```
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethodAsync Overload [▶ 895]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethodAsync Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRpcMethod> InvokeRpcMethodAsync(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String The symbol/Instance path of the symbol.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [▶ 1028] and the ErrorCode [▶ 992] of the ADS communication after execution.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethodAsync Overload [▶ 895]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.InvokeRpcMethodAsync Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRpcMethod> InvokeRpcMethodAsync(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The RPC callable symbol."
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethod [▶ 1025].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethod [▶ 1025] results contains the return value together with the output parameters.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.ReturnValue [▶ 1028] and the ErrorCode [▶ 992] of the ADS communication after execution.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

InvokeRpcMethodAsync Overload [▶ 895]

TwinCAT.Ads Namespace [▶ 151]

6.2.39.1.3 IAdsRpcInvoke.TryInvokeRpcMethod Method

Overload List

	Name	Description
	TryInvokeRpcMethod(String, String, .Object., Object.) [▶ 901]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .Object.., Object.) [▶ 903]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 905]	Invokes the rpc method.
	TryInvokeRpcMethod(IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 906]	Invokes the rpc method.

Reference

IAdsRpcInvoke Interface [▶ 886]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
```

```
    Object[] inParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});
```

```
        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

TryInvokeRpcMethod Overload [▶ 901]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol path.
methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

TryInvokeRpcMethod Overload [▶ 901]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.TryInvokeRpcMethod Method (String, String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryInvokeRpcMethod(
    string symbolPath,
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbolPath	Type: System.String The symbol.
methodName	Type: System.String Name of the method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
```

```
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

TryInvokeRpcMethod Overload [▶ 901]

TwinCAT.Ads Namespace [▶ 151]

IAdsRpcInvoke.TryInvokeRpcMethod Method (IRpcCallableInstance, IRpcMethod, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the rpc method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryInvokeRpcMethod(
    IRpcCallableInstance symbol,
    IRpcMethod method,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104] The RPC callable symbol
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
```

```
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IAdsRpcInvoke Interface [▶ 886]

TryInvokeRpcMethod Overload [▶ 901]

TwinCAT.Ads Namespace [▶ 151]

6.2.40 IAdsSession Interface

Interface IAdsSession

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSession : ISession,
    IConnectionStateProvider, ISymbolServerProvider
```

The IAdsSession type exposes the following members.

Properties

	Name	Description
	Address [▶ 910]	Gets the Address specifier of the Session / connection

	Name	Description
	AddressSpecifier [▶ 89]	Gets the communication endpoint address string representation. (Inherited from ISession [▶ 88].)
	Connection [▶ 90]	Gets the Connection object. (Inherited from ISession [▶ 88].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	EstablishedAt [▶ 90]	Gets the UTC time when the session was established. (Inherited from ISession [▶ 88].)
	Id [▶ 91]	Gets the Session Id (Inherited from ISession [▶ 88].)
	IsConnected [▶ 91]	Gets a value indicating whether the session is connected. (Inherited from ISession [▶ 88].)
	NetId [▶ 911]	Gets the NetId of the Session
	Owner [▶ 911]	Gets the Session owner.
	Port [▶ 911]	Gets the Ams Port of the Session
	Provider [▶ 91]	Gets the Session Provider (Inherited from ISession [▶ 88].)
	SymbolServer [▶ 100]	Gets the symbol server. (Inherited from ISymbolServerProvider [▶ 99].)

Methods

	Name	Description
	Close [▶ 92]	Closes this ISession [▶ 88] (Inherited from ISession [▶ 88].)
	Connect [▶ 93]	Connects the session and returns the established IConnection [▶ 74] object. (Inherited from ISession [▶ 88].)
	Disconnect [▶ 93]	Disconnects the ISession [▶ 88] (Inherited from ISession [▶ 88].)

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.ISession [▶ 88]

6.2.40.1 IAdsSession Properties

The IAdsSession [▶ 908] type exposes the following members.

Properties

	Name	Description
	Address [▶ 910]	Gets the Address specifier of the Session / connection
	AddressSpecifier [▶ 89]	Gets the communication endpoint address string representation. (Inherited from ISession [▶ 88].)
	Connection [▶ 90]	Gets the Connection object. (Inherited from ISession [▶ 88].)
	ConnectionState [▶ 85]	Gets the current Connection state of the IConnectionStateProvider [▶ 84] (Inherited from IConnectionStateProvider [▶ 84].)
	EstablishedAt [▶ 90]	Gets the UTC time when the session was established. (Inherited from ISession [▶ 88].)
	Id [▶ 91]	Gets the Session Id (Inherited from ISession [▶ 88].)
	IsConnected [▶ 91]	Gets a value indicating whether the session is connected. (Inherited from ISession [▶ 88].)
	NetId [▶ 911]	Gets the NetId of the Session
	Owner [▶ 911]	Gets the Session owner.
	Port [▶ 911]	Gets the Ams Port of the Session
	Provider [▶ 91]	Gets the Session Provider (Inherited from ISession [▶ 88].)
	SymbolServer [▶ 100]	Gets the symbol server. (Inherited from ISymbolServerProvider [▶ 99].)

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.1.1 IAdsSession.Address Property

Gets the Address specifier of the Session / connection

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The address.

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.1.2 IAdsSession.NetId Property

Gets the NetId of the Session

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The net identifier.

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.1.3 IAdsSession.Owner Property

Gets the Session owner.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object Owner { get; }
```

Property Value

Type: Object
The owner or NULL

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.1.4 IAdsSession.Port Property

Gets the Ams Port of the Session

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Port { get; }
```

Property Value

Type: Int32
The port.

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.2 IAdsSession Methods

The IAdsSession [▶ 908] type exposes the following members.

Methods

	Name	Description
	Close [▶ 92]	Closes this ISession [▶ 88] (Inherited from ISession [▶ 88].)
	Connect [▶ 93]	Connects the session and returns the established IConnection [▶ 74] object. (Inherited from ISession [▶ 88].)
	Disconnect [▶ 93]	Disconnects the ISession [▶ 88] (Inherited from ISession [▶ 88].)

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.40.3 IAdsSession Events

The IAdsSession [▶ 908] type exposes the following members.

Events

	Name	Description
	ConnectionStateChanged [▶ 86]	Occurs when connection status of the IConnectionStateProvider [▶ 84] has been changed. (Inherited from IConnectionStateProvider [▶ 84].)

Reference

IAdsSession Interface [▶ 908]

TwinCAT.Ads Namespace [▶ 151]

6.2.41 IAdsSessionSettings Interface

Interface for ADS Session Settings

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSessionSettings
```

The IAdsSessionSettings type exposes the following members.

Properties

	Name	Description
	ResurrectionTime [▶ 913]	Gets or sets the resurrection time.
	SymbolLoader [▶ 914]	Gets or sets the symbol loader settings
	Timeout [▶ 914]	Gets the ADS timeout in milliseconds.

Reference

TwinCAT.Ads Namespace [▶ 151]

ISessionSettings

6.2.41.1 IAdsSessionSettings Properties

The IAdsSessionSettings [▶ 913] type exposes the following members.

Properties

	Name	Description
	ResurrectionTime [▶ 913]	Gets or sets the resurrection time.
	SymbolLoader [▶ 914]	Gets or sets the symbol loader settings
	Timeout [▶ 914]	Gets the ADS timeout in milliseconds.

Reference

IAdsSessionSettings Interface [▶ 913]

TwinCAT.Ads Namespace [▶ 151]

6.2.41.1.1 IAdsSessionSettings.ResurrectionTime Property

Gets or sets the resurrection time.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TimeSpan ResurrectionTime { get; set; }
```

Property Value

Type: TimeSpan
The resurrection time.

Reference

IAdsSessionSettings Interface [▶ 913]

TwinCAT.Ads Namespace [▶ 151]

6.2.41.1.2 IAdsSessionSettings.SymbolLoader Property

Gets or sets the symbol loader settings

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
SymbolLoaderSettings SymbolLoader { get; set; }
```

Property Value

Type: SymbolLoaderSettings [▶ 140]
The symbol loader.

Reference

IAdsSessionSettings Interface [▶ 913]

TwinCAT.Ads Namespace [▶ 151]

6.2.41.1.3 IAdsSessionSettings.Timeout Property

Gets the ADS timeout in milliseconds.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Timeout { get; }
```

Property Value

Type: Int32
The timeout.

Reference

IAdsSessionSettings Interface [▶ 913]

TwinCAT.Ads Namespace [▶ 151]

6.2.42 IAdsStateControl Interface

Interface for reading and controlling the ADS state.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsStateControl
```

Methods

	Name	Description
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.42.1 IAdsStateControl Methods

Methods

	Name	Description
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	

Reference

IAdsStateControl Interface [▶ 915]

TwinCAT.Ads Namespace [▶ 151]

6.2.42.1.1 IAdsStateControl.TryWriteControl Method

Overload List

	Name	Description
	TryWriteControl(StateInfo) [▶ 917]	Changes the ADS status and the device status of an ADS server.

	Name	Description
	TryWriteControl(StateInfo, ReadOnlyMemory) [▶ 917]	

Reference

IAdsStateControl Interface [▶ 915]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.TryWriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsStateControl Interface [▶ 915]

TryWriteControl Overload [▶ 916]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.TryWriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsStateControl Interface [▶ 915]

TryWriteControl Overload [▶ 916]

TwinCAT.Ads Namespace [▶ 151]

6.2.42.1.2 IAdsStateControl.WriteControl Method

Overload List

	Name	Description
	WriteControl(StateInfo) [▶ 918]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory) [▶ 919]	

Reference

IAdsStateControl Interface [▶ 915]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.WriteControl Method (StateInfo)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteControl(
    StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

Reference

IAdsStateControl Interface [▶ 915]

WriteControl Overload [▶ 918]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.WriteControl Method (StateInfo, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

Reference

IAdsStateControl Interface [▶ 915]

WriteControl Overload [▶ 918]

TwinCAT.Ads Namespace [▶ 151]

6.2.42.1.3 IAdsStateControl.WriteControlAsync Method

Overload List

	Name	Description
	WriteControlAsync(AdsState, UInt16, CancellationToken) [▶ 920]	Changes the ADS status and device status of the ADS server asynchronously.
	WriteControlAsync(AdsState, UInt16, ReadOnlyMemory, Void) [▶ 921]	

Reference

IAdsStateControl Interface [▶ 915]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.WriteControlAsync Method (AdsState, UInt16, CancellationToken)

Changes the ADS status and device status of the ADS server asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

state	Type: TwinCAT.Ads.AdsState [▶ 626] The ADS state.
deviceState	Type: System.UInt16 The device state.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'WriteControl' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsStateControl Interface [▶ 915]

WriteControlAsync Overload [▶ 920]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControl.WriteControlAsync Method (AdsState, UInt16, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> WriteControlAsync(
    AdsState state,
    ushort deviceState,
    ReadOnlyMemory data,
    void cancel
)
```

Parameters

state	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
data	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultAds [▶ 989].

Reference

IAdsStateControl Interface [▶ 915]

WriteControlAsync Overload [▶ 920]

TwinCAT.Ads Namespace [▶ 151]

6.2.43 IAdsStateControlTimeout Interface

Interface IAdsStateControlTimeout

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsStateControlTimeout
```

The IAdsStateControlTimeout type exposes the following members.

Methods

	Name	Description
	ReadState [▶ 923]	Reads the ADS status and the device status from an ADS server.
	TryReadState [▶ 924]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryWriteControl(StateInfo, Int32) [▶ 925]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 925]	
	WriteControl(StateInfo, Int32) [▶ 926]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 927]	

Remarks

For internal use only.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.43.1 IAdsStateControlTimeout Methods

The IAdsStateControlTimeout [▶ 921] type exposes the following members.

Methods

	Name	Description
	ReadState [▶ 923]	Reads the ADS status and the device status from an ADS server.
	TryReadState [▶ 924]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	TryWriteControl(StateInfo, Int32) [▶ 925]	Changes the ADS status and the device status of an ADS server.

	Name	Description
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 925]	
	WriteControl(StateInfo, Int32) [▶ 926]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 927]	

Reference

IAdsStateControlTimeout Interface [▶ 921]

TwinCAT.Ads Namespace [▶ 151]

6.2.43.1.1 IAdsStateControlTimeout.ReadState Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
StateInfo ReadState(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout.

Return Value

Type: StateInfo [▶ 1041]
The ADS statue and device status.

Reference

IAdsStateControlTimeout Interface [▶ 921]

TwinCAT.Ads Namespace [▶ 151]

6.2.43.1.2 IAdsStateControlTimeout.TryReadState Method

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadState(
    int timeout,
    out StateInfo stateInfo
)
```

Parameters

timeout	Type: System.Int32 The timeout.
stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode of the ADS read state call. Check for AdsErrorCode.NoError to see if call was successful.

Reference

IAdsStateControlTimeout Interface [▶ 921]

TwinCAT.Ads Namespace [▶ 151]

6.2.43.1.3 IAdsStateControlTimeout.TryWriteControl Method

Overload List

	Name	Description
	TryWriteControl(StateInfo, Int32) [▶ 925]	Changes the ADS status and the device status of an ADS server.
	TryWriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 925]	

Reference

IAdsStateControlTimeout Interface [▶ 921]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControlTimeout.TryWriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    int timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.
timeout	Type: System.Int32 The timeout.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsStateControlTimeout Interface [▶ 921]

TryWriteControl Overload [▶ 924]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControlTimeout.TryWriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory

timeout	Type: System.Void

Return Value

Type: AdsErrorCode [▶ 575]

Reference

IAdsStateControlTimeout Interface [▶ 921]

TryWriteControl Overload [▶ 924]

TwinCAT.Ads Namespace [▶ 151]

6.2.43.1.4 IAdsStateControlTimeout.WriteControl Method

Overload List

	Name	Description
	WriteControl(StateInfo, Int32) [▶ 926]	Changes the ADS status and the device status of an ADS server.
	WriteControl(StateInfo, ReadOnlyMemory, Void) [▶ 927]	

Reference

IAdsStateControlTimeout Interface [▶ 921]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControlTimeout.WriteControl Method (StateInfo, Int32)

Changes the ADS status and the device status of an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteControl(
    StateInfo stateInfo,
    int timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041] New ADS status and device status.

timeout	Type: System.Int32 The timeout.

Reference

IAdsStateControlTimeout Interface [▶ 921]

WriteControl Overload [▶ 926]

TwinCAT.Ads Namespace [▶ 151]

IAdsStateControlTimeout.WriteControl Method (StateInfo, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteControl(
    StateInfo stateInfo,
    ReadOnlyMemory writeBuffer,
    void timeout
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]
writeBuffer	Type: ReadOnlyMemory
timeout	Type: System.Void

Reference

IAdsStateControlTimeout Interface [▶ 921]

WriteControl Overload [▶ 926]

TwinCAT.Ads Namespace [▶ 151]

6.2.44 IAdsStateObserver Interface

Interface for an AdsState observer

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsStateObserver
```

The IAdsStateObserver type exposes the following members.

Properties

	Name	Description
	StateInfo [▶ 928]	Gets the current state of the connected ADS Server.

Events

	Name	Description
	AdsStateChanged [▶ 929]	Occurs when the ads state has been changed.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.44.1 IAdsStateObserver Properties

The IAdsStateObserver [▶ 927] type exposes the following members.

Properties

	Name	Description
	StateInfo [▶ 928]	Gets the current state of the connected ADS Server.

Reference

IAdsStateObserver Interface [▶ 927]

TwinCAT.Ads Namespace [▶ 151]

6.2.44.1.1 IAdsStateObserver.StateInfo Property

Gets the current state of the connected ADS Server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
StateInfo StateInfo { get; }
```

Property Value

Type: StateInfo [▶ 1041]
ADS state

Reference

IAdsStateObserver Interface [▶ 927]

TwinCAT.Ads Namespace [▶ 151]

6.2.44.2 IAdsStateObserver Events

The IAdsStateObserver [▶ 927] type exposes the following members.

Events

	Name	Description
⚡	AdsStateChanged [▶ 929]	Occurs when the ads state has been changed.

Reference

IAdsStateObserver Interface [▶ 927]

TwinCAT.Ads Namespace [▶ 151]

6.2.44.2.1 IAdsStateObserver.AdsStateChanged Event

Occurs when the ads state has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsStateChangedEventArgs2> AdsStateChanged
```

Value

Type: System.EventHandler.AdsStateChangedEventArgs2 [▶ 629].

Reference

IAdsStateObserver Interface [▶ 927]

TwinCAT.Ads Namespace [▶ 151]

6.2.45 IAdsStateProvider Interface

Interface IAdsStateProvider

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsStateProvider
```

The IAdsStateProvider type exposes the following members.

Methods

	Name	Description
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server.
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.

Events

	Name	Description
	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1 IAdsStateProvider Methods

The IAdsStateProvider [▶ 929] type exposes the following members.

Methods

	Name	Description
	ReadState [▶ 931]	Reads the ADS status and the device status from an ADS server.
	ReadStateAsync [▶ 931]	Reads the ADS status and the device status from an ADS server.
	RegisterAdsStateChangedAsync [▶ 932]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.
	TryReadState [▶ 932]	Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.
	UnregisterAdsStateChangedAsync [▶ 933]	Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1.1 IAdsStateProvider.ReadState Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
StateInfo ReadState()
```

Return Value

Type: StateInfo [▶ 1041]
The ADS statue and device status.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1.2 IAdsStateProvider.ReadStateAsync Method

Reads the ADS status and the device status from an ADS server.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadDeviceState> ReadStateAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultReadDeviceState [▶ 1016].
A task that represents the asynchronous 'ReadState' operation. The ResultReadDeviceState [▶ 1016] parameter contains the state (State [▶ 1018]) as long as the ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1.3 IAdsStateProvider.RegisterAdsStateChangedAsync Method

Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> RegisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be registered for AdsStateChanged calls.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1.4 IAdsStateProvider.TryReadState Method

Reads the ADS status and the device status from an ADS server. Unlike the ReadState method this method does not call an exception on failure. Instead an AdsErrorCode is returned. If the return value is equal to AdsErrorCode.NoError the call was successful.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadState(
    out StateInfo stateInfo
)
```

Parameters

stateInfo	Type: TwinCAT.Ads.StateInfo [▶ 1041]. The ADS statue and device status.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode [▶ 575] of the ADS read state call. Check for NoError [▶ 575] to see if call was successful.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.1.5 IAdsStateProvider.UnregisterAdsStateChangedAsync Method

Registers for AdsStateChanged [▶ 934] events as an asynchronous operation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> UnregisterAdsStateChangedAsync(
    EventHandler<AdsStateChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627]. The handler function to be unregistered.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterAdsStateChanged' operation. The ResultAds [▶ 989] parameter contains the state the ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.2 IAdsStateProvider Events

The IAdsStateProvider [▶ 929] type exposes the following members.

Events

	Name	Description
⚡	AdsStateChanged [▶ 934]	Occurs when the AdsState of the target system has been changed.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.45.2.1 IAdsStateProvider.AdsStateChanged Event

Occurs when the AdsState of the target system has been changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsStateChangedEventArgs> AdsStateChanged
```

Value

Type: System.EventHandler.AdsStateChangedEventArgs [▶ 627].

Remarks

This event occurs asynchronously if the synchronized flag is not set.

Reference

IAdsStateProvider Interface [▶ 929]

TwinCAT.Ads Namespace [▶ 151]

6.2.46 IAdsSymbolChangedProvider Interface

Interface IAdsConnectionLegacy

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSymbolChangedProvider
```

The IAdsSymbolChangedProvider type exposes the following members.

Methods

	Name	Description
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously.
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous.

Events

	Name	Description
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.46.1 IAdsSymbolChangedProvider Methods

The IAdsSymbolChangedProvider [▶ 934] type exposes the following members.

Methods

	Name	Description
	RegisterSymbolVersionChangedAsync [▶ 935]	Registers the symbol version changed asynchronously.
	UnregisterSymbolVersionChangedAsync [▶ 936]	Unregisters the symbol version changed asynchronous.

Reference

IAdsSymbolChangedProvider Interface [▶ 934]

TwinCAT.Ads Namespace [▶ 151]

6.2.46.1.1 IAdsSymbolChangedProvider.RegisterSymbolVersionChangedAsync Method

Registers the symbol version changed asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> RegisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to register.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsSymbolChangedProvider Interface [▶ 934]

TwinCAT.Ads Namespace [▶ 151]

6.2.46.1.2 IAdsSymbolChangedProvider.UnregisterSymbolVersionChangedAsync Method

Unregisters the symbol version changed asynchronous.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAds> UnregisterSymbolVersionChangedAsync(
    EventHandler<AdsSymbolVersionChangedEventArgs> handler,
    CancellationToken cancel
)
```

Parameters

handler	Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637]. The handler function to unregister.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAds [▶ 989].
A task that represents the asynchronous 'UnregisterSymbolVersionChangedAsync' operation. The ResultAds [▶ 989] parameter contains the value ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsSymbolChangedProvider Interface [▶ 934]

TwinCAT.Ads Namespace [▶ 151]

6.2.46.2 IAdsSymbolChangedProvider Events

The IAdsSymbolChangedProvider [▶ 934] type exposes the following members.

Events

	Name	Description
	AdsSymbolVersionChanged [▶ 937]	Occurs when the ADS Symbol Version changed.

Reference

IAdsSymbolChangedProvider Interface [▶ 934]

TwinCAT.Ads Namespace [▶ 151]

6.2.46.2.1 IAdsSymbolChangedProvider.AdsSymbolVersionChanged Event

Occurs when the ADS Symbol Version changed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AdsSymbolVersionChangedEventArgs> AdsSymbolVersionChanged
```

Value

Type: System.EventHandler.AdsSymbolVersionChangedEventArgs [▶ 637].

Reference

IAdsSymbolChangedProvider Interface [▶ 934]

TwinCAT.Ads Namespace [▶ 151]

6.2.47 IAdsSymbolicAccess Interface

Interface for symbolic ads access.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSymbolicAccess
```

The IAdsSymbolicAccess type exposes the following members.

Methods

	Name	Description
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache.
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type.
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type.
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 950]	Reads the value of a symbol asynchronously and returns it as an object.
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type.
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices.

	Name	Description
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object.
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object.
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.47.1 IAdsSymbolicAccess Methods

The IAdsSymbolicAccess [▶ 937] type exposes the following members.

Methods

	Name	Description
	CleanupSymbolTable [▶ 942]	Clears the internal symbol / DataTypes cache.
	ReadDataType [▶ 942]	Call this method to obtain information about the specified data type.
	ReadDataTypeAsync [▶ 943]	Call this method to obtain information about the specified data type.
	ReadSymbol [▶ 943]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadSymbolAsync [▶ 944]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.
	ReadValue.T.(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously.
	ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 950]	Reads the value of a symbol asynchronously and returns it as an object.
	TryReadDataType [▶ 952]	Call this method to obtain information about the specified data type.
	TryReadSymbol [▶ 953]	Call this method to obtain information about the individual symbols (variables) in ADS devices.
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object.

	Name	Description
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object.
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object.
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValue.T.(ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

6.2.47.1.1 IAdsSymbolicAccess.CleanupSymbolTable Method

Clears the internal symbol / DataTypes cache.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void CleanupSymbolTable()
```

Remarks

Previously stored symbol information is cleared. As a consequence the symbol information must be obtained from the ADS server again if accessed, which which needs an extra ADS round trip.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

6.2.47.1.2 IAdsSymbolicAccess.ReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ReadDataType(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the data type (without namespace)

Return Value

Type: IDataType [▶ 1986]
An containing the requested type.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

IAdsSymbolicAccess.ReadDataTypeAsync(String, CancellationToken) [▶ 943]

6.2.47.1.3 IAdsSymbolicAccess.ReadDataTypeAsync Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<IDataType>> ReadDataTypeAsync(
    string typeName,
    CancellationToken cancel
)
```

Parameters

typeName	Type: System.String Name of the data type.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IDataType [▶ 1986]..
A task that represents the asynchronous 'ReadDataType' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

IAdsSymbolicAccess.TryReadDataType(String, IDataType.) [▶ 952]

6.2.47.1.4 IAdsSymbolicAccess.ReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IAdsSymbol ReadSymbol(
    string symbolPath
)
```

Parameters

symbolPath	Type: System.String Name of the symbol.

Return Value

Type: IAdsSymbol [▶ 1379]
An IAdsSymbol [▶ 1379] containing the requested symbol information or null if symbol could not be found.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol.) [▶ 953]

IAdsSymbolicAccess.ReadSymbolAsync(String, CancellationToken) [▶ 944]

6.2.47.1.5 IAdsSymbolicAccess.ReadSymbolAsync Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<IAdsSymbol>> ReadSymbolAsync(
    string symbolPath,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String Name of the symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultValue [▶ 1029].IAdsSymbol [▶ 1379]..
A task that represents the asynchronous 'ReadSymbolInfo' operation. The ResultValue.TValue. [▶ 1029] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadSymbol(String) [▶ 943]

IAdsSymbolicAccess.TryReadSymbol(String, IAdsSymbol.) [▶ 953]

6.2.47.1.6 IAdsSymbolicAccess.ReadValue Method

Overload List

	Name	Description
	ReadValue.T.(String) [▶ 945]	Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.
	ReadValue(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValue.T.(ISymbol) [▶ 946]	Reads the value of a symbol and returns it as an object.
	ReadValue(String, Type) [▶ 947]	Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue.T. Method (String)

Reads the value of a symbol and returns the value. The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadValue<T>(
    string symbolPath
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.

Type Parameters

T	The value type

Return Value

Type: T
Value of the symbol

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValue Overload [▶ 945]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadValue.T.(String, T.) [▶ 954]

IAdsSymbolicAccess.ReadValueAsync.T.(String, CancellationToken) [▶ 948]

IAdsSymbolicAccess.ReadValue Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadValue(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Return Value

Type: Object
The value of the symbol as an object.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValue Overload [▶ 945]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [▶ 954]

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [▶ 949]

IAdsSymbolicAccess.ReadValue.T. Method (ISymbol)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T ReadValue<T>(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.

Type Parameters

T	The value type.

Return Value

Type: T
The value of the symbol.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValue Overload [▶ 945]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.) [▶ 955]

IAdsSymbolicAccess.ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 950]

IAdsSymbolicAccess.ReadValue Method (String, Type)

Reads the value of a symbol and returns the value as object. The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadValue(
    string symbolPath,
    Type type
)
```

Parameters

symbolPath	Type: System.String Symbol Path of the ADS symbol.

type	Type: System.Type Managed type of the ADS symbol.

Return Value

Type: Object
Value of the symbol

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValue Overload [▶ 945]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadValue(String, Type, Object.) [▶ 956]

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [▶ 951]

6.2.47.1.7 IAdsSymbolicAccess.ReadValueAsync Method

Overload List

	Name	Description
	ReadValueAsync.T.(String, CancellationToken) [▶ 948]	Reads the value of a symbol asynchronously.
	ReadValueAsync(ISymbol, CancellationToken) [▶ 949]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync.T.(ISymbol, CancellationToken) [▶ 950]	Reads the value of a symbol asynchronously and returns it as an object.
	ReadValueAsync(String, Type, CancellationToken) [▶ 951]	Reads the value of a symbol asynchronously.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValueAsync.T. Method (String, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadValueAsync<T>(
    string symbolPath,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Remarks

The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValueAsync Overload [▶ 948]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue.T.(String) [▶ 945]

IAdsSymbolicAccess.TryReadValue.T.(String, T.) [▶ 954]

IAdsSymbolicAccess.ReadValueAsync Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadValueAsync(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'..

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValueAsync Overload [▶ 948]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue(ISymbol) [▶ 946]

IAdsSymbolicAccess.ReadValue(ISymbol) [▶ 946]

IAdsSymbolicAccess.TryReadValue(ISymbol, Object.) [▶ 954]

IAdsSymbolicAccess.ReadValueAsync.T. Method (ISymbol, CancellationToken)

Reads the value of a symbol asynchronously and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<T>> ReadValueAsync<T>(
    ISymbol symbol,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultValue [▶ 1029].T..
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1032]) and the ErrorCode [▶ 992] after execution.

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValueAsync Overload [▶ 948]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue.T.(ISymbol) [▶ 946]

IAdsSymbolicAccess.TryReadValue.T.(ISymbol, T.) [▶ 955]

IAdsSymbolicAccess.ReadValueAsync Method (String, Type, CancellationToken)

Reads the value of a symbol asynchronously.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAnyValue> ReadValueAsync(
    string symbolPath,
    Type type,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Return Value

Type: Task.ResultAnyValue [▶ 998].
A task that represents the asynchronous read operation. The ResultAnyValue [▶ 998] parameter contains the read value (Value [▶ 1000]) and the ErrorCode [▶ 992] after execution.

Remarks

The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

ReadValueAsync Overload [▶ 948]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue(String, Type) [▶ 947]

IAdsSymbolicAccess.TryReadValue(String, Type, Object.) [▶ 956]

6.2.47.1.8 IAdsSymbolicAccess.TryReadDataType Method

Call this method to obtain information about the specified data type.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadDataType(
    string typeName,
    out IDataType dataType
)
```

Parameters

typeName	Type: System.String Name of the symbol.
dataType	Type: TwinCAT.TypeSystem.IDataType [▶ 1986]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
A IDataType [▶ 1986] containing the requested symbol information or null if symbol could not be found.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadDataType(String) [▶ 942]

6.2.47.1.9 IAdsSymbolicAccess.TryReadSymbol Method

Call this method to obtain information about the individual symbols (variables) in ADS devices.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadSymbol(
    string symbolPath,
    out IAdsSymbol symbol
)
```

Parameters

symbolPath	Type: System.String Name of the symbol.
symbol	Type: TwinCAT.Ads.TypeSystem.IAdsSymbol [▶ 1379]. The symbol.

Return Value

Type: AdsErrorCode [▶ 575]
An IAdsSymbol [▶ 1379] containing the requested symbol information or null if symbol could not be found.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadSymbol(String) [▶ 943]

6.2.47.1.10 IAdsSymbolicAccess.TryReadValue Method

Overload List

	Name	Description
	TryReadValue.T.(String, T.) [▶ 954]	Reads the value of a symbol and returns the value as object.
	TryReadValue(ISymbol, Object.) [▶ 954]	Reads the value of a symbol and returns it as an object.
	TryReadValue.T.(ISymbol, T.) [▶ 955]	Reads the value of a symbol and returns it as an object.
	TryReadValue(String, Type, Object.) [▶ 956]	Reads the value of a symbol and returns the value as object.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryReadValue.T. Method (String, T.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadValue<T>(
    string symbolPath,
    out T value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: T. The read value of the Symbol.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Remarks

The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryReadValue Overload [▶ 953]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue.T.(String) [▶ 945]

IAdsSymbolicAccess.ReadValueAsync.T.(String, CancellationToken) [▶ 948]

IAdsSymbolicAccess.TryReadValue Method (ISymbol, Object.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadValue(
    ISymbol symbol,
    out Object value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: System.Object. The value.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryReadValue Overload [▶ 953]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue(ISymbol) [▶ 946]

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [▶ 949]

IAdsSymbolicAccess.TryReadValue.T. Method (ISymbol, T.)

Reads the value of a symbol and returns it as an object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadValue<T>(
    ISymbol symbol,
    out T value
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol that should be read.
value	Type: T. The value.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code

Remarks

Supported types are limited to 'AnyTypes', what includes all primitive types (UInt32, Int32, Bool etc.), strings, and Arrays that are compositions of 'AnyTypes'.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryReadValue Overload [▶ 953]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue.T.(ISymbol) [▶ 946]

IAdsSymbolicAccess.ReadValueAsync(ISymbol, CancellationToken) [▶ 949]

IAdsSymbolicAccess.TryReadValue Method (String, Type, Object.)

Reads the value of a symbol and returns the value as object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryReadValue(
    string symbolPath,
    Type type,
    out Object value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
type	Type: System.Type Managed type of the ADS symbol.
value	Type: System.Object. The read value of the Symbol.

Return Value

Type: AdsErrorCode [▶ 575]
The AdsErrorCode [▶ 575].

Remarks

The parameter type must have the same binary layout as the ADS symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryReadValue Overload [▶ 953]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.ReadValue(String, Type) [▶ 947]

IAdsSymbolicAccess.ReadValueAsync.T.(String, CancellationToken) [▶ 948]

IAdsSymbolicAccess.ReadValueAsync(String, Type, CancellationToken) [▶ 951]

6.2.47.1.11 IAdsSymbolicAccess.TryWriteValue Method

Overload List

	Name	Description
	TryWriteValue(String, Object) [▶ 957]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.
	TryWriteValue.T.(String, T) [▶ 958]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	TryWriteValue(ISymbol, Object) [▶ 959]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	TryWriteValue.T.(ISymbol, T) [▶ 960]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryWriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteValue(
    string symbolPath,
    Object value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryWriteValue Overload [▶ 957]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue(String, Object) [▶ 961]

IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]

IAdsSymbolicAccess.TryWriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteValue<T>(
    string symbolPath,
    T value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryWriteValue Overload [▶ 957]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue.T.(String, T) [▶ 961]

IAdsSymbolicAccess.TryWriteValue.T.(String, T)

IAdsSymbolicAccess.TryWriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryWriteValue Overload [▶ 957]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [▶ 962]

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]

IAdsSymbolicAccess.TryWriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryWriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TryWriteValue Overload [▶ 957]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue.T.(ISymbol, T) [▶ 963]

IAdsSymbolicAccess.WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]

6.2.47.1.12 IAdsSymbolicAccess.WriteValue Method

Overload List

	Name	Description
	WriteValue(String, Object) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValue.T.(String, T) [▶ 961]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.

	Name	Description
	WriteValue(ISymbol, Object) [▶ 962]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValue.T. (ISymbol, T) [▶ 963]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue Method (String, Object)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue(
    string symbolPath,
    Object value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: System.Object Object holding the value to be written to the ADS symbol

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValue Overload [▶ 960]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryWriteValue(String, Object) [▶ 957]

IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]

IAdsSymbolicAccess.WriteValue.T. Method (String, T)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue<T>(
    string symbolPath,
    T value
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol

Type Parameters

T	the value type.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValue Overload [▶ 960]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryWriteValue.T.(String, T) [▶ 958]

IAdsSymbolicAccess.WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]

IAdsSymbolicAccess.WriteValue Method (ISymbol, Object)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue(
    ISymbol symbol,
    Object val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValue Overload [▶ 960]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [▶ 959]

IAdsSymbolicAccess.WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]

IAdsSymbolicAccess.WriteValue.T. Method (ISymbol, T)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue<T>(
    ISymbol symbol,
    T val
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.

Type Parameters

T	The value type.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValue Overload [▶ 960]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [▶ 960]

IAdsSymbolicAccess.WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]

6.2.47.1.13 IAdsSymbolicAccess.WriteValueAsync Method

Overload List

	Name	Description
	WriteValueAsync.T.(String, T, CancellationToken) [▶ 964]	Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.
	WriteValueAsync(ISymbol, Object, CancellationToken) [▶ 965]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.
	WriteValueAsync.T.(ISymbol, T, CancellationToken) [▶ 966]	Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Reference

IAdsSymbolicAccess Interface [▶ 937]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValueAsync.T. Method (String, T, CancellationToken)

Writes the passed object value to the specified ADS symbol.The parameter type must have the same binary layout as the ADS symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteValueAsync<T>(
    string symbolPath,
    T value,
    CancellationToken cancel
)
```

Parameters

symbolPath	Type: System.String Name of the ADS symbol.
value	Type: T Object holding the value to be written to the ADS symbol
cancel	Type: System.Threading.CancellationToken The cancel token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValueAsync Overload [▶ 964]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue.T.(String, T) [▶ 961]

IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [▶ 960]

IAdsSymbolicAccess.WriteValueAsync Method (ISymbol, Object, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteValueAsync(
    ISymbol symbol,
    Object val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: System.Object The value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValueAsync Overload [▶ 964]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue(ISymbol, Object) [▶ 962]

IAdsSymbolicAccess.TryWriteValue(ISymbol, Object) [▶ 959]

IAdsSymbolicAccess.WriteValueAsync.T. Method (ISymbol, T, CancellationToken)

Writes a value to the symbol. Strings and all primitive data types(UInt32, Int32, Bool etc.) are supported. If a string is passed as parameter, the method attempts to parse the string according to the ADS data type of the symbol.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWrite> WriteValueAsync<T>(
    ISymbol symbol,
    T val,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol the value is written to.
val	Type: T The value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The value type.

Return Value

Type: Task.ResultWrite [▶ 1032].
A task that represents the asynchronous 'WriteSymbol' operation. The ResultWrite [▶ 1032] parameter contains the ErrorCode [▶ 992] after execution.

Reference

IAdsSymbolicAccess Interface [▶ 937]

WriteValueAsync Overload [▶ 964]

TwinCAT.Ads Namespace [▶ 151]

IAdsSymbolicAccess.WriteValue.T.(ISymbol, T) [▶ 963]

IAdsSymbolicAccess.TryWriteValue.T.(ISymbol, T) [▶ 960]

6.2.48 IAdsSymbolTableProvider Interface

Interface IAdsSymbolTableProvider

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSymbolTableProvider
```

The IAdsSymbolTableProvider type exposes the following members.

Methods

	Name	Description
	GetSymbolTableAsync [▶ 967]	Gets the symbol table asynchronously
	SetSymbolEncoding [▶ 968]	Sets the default encoding.
	TryGetSymbolTable [▶ 968]	Get the symbol table.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.48.1 IAdsSymbolTableProvider Methods

The IAdsSymbolTableProvider [▶ 967] type exposes the following members.

Methods

	Name	Description
	GetSymbolTableAsync [▶ 967]	Gets the symbol table asynchronously
	SetSymbolEncoding [▶ 968]	Sets the default encoding.
	TryGetSymbolTable [▶ 968]	Get the symbol table.

Reference

IAdsSymbolTableProvider Interface [▶ 967]

TwinCAT.Ads Namespace [▶ 151]

6.2.48.1.1 IAdsSymbolTableProvider.GetSymbolTableAsync Method

Gets the symbol table asynchronously

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<ISymbolInfoTable>> GetSymbolTableAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultValue [▶ 1029].ISymbolInfoTable..
A task that represents the asynchronous 'RegisterSymbolVersionChanged' operation. The ResultValue.TValue. [▶ 1029] parameter contains the value Value [▶ 1032] and the return code ErrorCode [▶ 992] of the ADS communication after execution.

Reference

IAdsSymbolTableProvider Interface [▶ 967]

TwinCAT.Ads Namespace [▶ 151]

6.2.48.1.2 IAdsSymbolTableProvider.SetSymbolEncoding Method

Sets the default encoding.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void SetSymbolEncoding(
    Encoding encoding
)
```

Parameters

encoding	Type: System.Text.Encoding The encoding.

Reference

IAdsSymbolTableProvider Interface [▶ 967]

TwinCAT.Ads Namespace [▶ 151]

6.2.48.1.3 IAdsSymbolTableProvider.TryGetSymbolTable Method

Get the symbol table.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode TryGetSymbolTable(
    out ISymbolInfoTable table
)
```

Parameters

table	Type: ISymbolInfoTable. The symbol table.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS Error Code.

Reference

IAdsSymbolTableProvider Interface [▶ 967]

TwinCAT.Ads Namespace [▶ 151]

6.2.49 INotification Interface

Common INotification interface

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface INotification
```

The INotification type exposes the following members.

Properties

	Name	Description
	Data [▶ 970]	The notification Data.
	Handle [▶ 970]	The notification handle
	TimeStamp [▶ 971]	Gets the time stamp of the INotification
	UserData [▶ 971]	Attached UserData/Tag at the INotification

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.49.1 INotification Properties

The INotification [▶ 969] type exposes the following members.

Properties

	Name	Description
	Data [▶ 970]	The notification Data.
	Handle [▶ 970]	The notification handle
	TimeStamp [▶ 971]	Gets the time stamp of the INotification [▶ 969]
	UserData [▶ 971]	Attached UserData/Tag at the INotification [▶ 969]

Reference

INotification Interface [▶ 969]

TwinCAT.Ads Namespace [▶ 151]

6.2.49.1.1 INotification.Data Property

The notification Data.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ReadOnlyMemory Data { get; }
```

Property Value

Type: ReadOnlyMemory
The data.

Reference

INotification Interface [▶ 969]

TwinCAT.Ads Namespace [▶ 151]

6.2.49.1.2 INotification.Handle Property

The notification handle

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint Handle { get; }
```

Property Value

Type: UInt32
The handle.

Remarks

This can be the Client handle or the Server handle!

Reference

INotification Interface [▶ 969]

TwinCAT.Ads Namespace [▶ 151]

6.2.49.1.3 INotification.TimeStamp Property

Gets the time stamp of the INotification [▶ 969]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The time stamp.

Reference

INotification Interface [▶ 969]

TwinCAT.Ads Namespace [▶ 151]

6.2.49.1.4 INotification.UserData Property

Attached UserData/Tag at the INotification [▶ 969]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object UserData { get; }
```

Property Value

Type: Object
The user data.

Reference

INotification Interface [▶ 969]

TwinCAT.Ads Namespace [▶ 151]

6.2.50 INotificationSettings Interface

Interface for Notification Settings Implements the IComparable.T.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface INotificationSettings : IComparable<INotificationSettings>
```

The INotificationSettings type exposes the following members.

Methods

	Name	Description
	CompareTo	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. (Inherited from IComparable.INotificationSettings..)

Reference

TwinCAT.Ads Namespace [▶ 151]

System.IComparable.T.

6.2.50.1 INotificationSettings Methods

The INotificationSettings [▶ 972] type exposes the following members.

Methods

	Name	Description
	CompareTo	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. (Inherited from IComparable.INotificationSettings [▶ 972]..)

Reference

INotificationSettings Interface [▶ 972]

TwinCAT.Ads Namespace [▶ 151]

6.2.51 IRouterNotificationProvider Interface

Interface for AMS Router Notifications.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRouterNotificationProvider
```

The IRouterNotificationProvider type exposes the following members.

Events

	Name	Description
⚡	RouterStateChanged [▶ 973]	Router state changed event.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.51.1 IRouterNotificationProvider Events

The IRouterNotificationProvider [▶ 973] type exposes the following members.

Events

	Name	Description
⚡	RouterStateChanged [▶ 973]	Router state changed event.

Reference

IRouterNotificationProvider Interface [▶ 973]

TwinCAT.Ads Namespace [▶ 151]

6.2.51.1.1 IRouterNotificationProvider.RouterStateChanged Event

Router state changed event.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<AmsRouterNotificationEventArgs> RouterStateChanged
```

Value

Type: System.EventHandler.AmsRouterNotificationEventArgs [▶ 695].

Reference

IRouterNotificationProvider Interface [▶ 973]

TwinCAT.Ads Namespace [▶ 151]

6.2.52 Notification Class

Class Notification. Implements the INotification [▶ 969]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Notification
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Notification : INotification
```

The Notification type exposes the following members.

Constructors

	Name	Description
	Notification(INotification) [▶ 975]	Initializes a new instance of the Notification class.
	Notification(UInt32, DateTimeOffset, Object, ReadOnlyMemory) [▶ 976]	

Properties

	Name	Description
	Data [▶ 977]	The notification Data.
	Handle [▶ 977]	The notification handle
	TimeStamp [▶ 978]	Gets the time stamp of the INotification [▶ 969]
	UserData [▶ 978]	Attached UserData/Tag at the INotification [▶ 969]

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.INotification [▶ 969]

6.2.52.1 Notification Constructor

Overload List

	Name	Description
	Notification(INotification) [▶ 975]	Initializes a new instance of the Notification [▶ 974] class.
	Notification(UInt32, DateTimeOffset, Object, ReadOnlyMemory) [▶ 976]	

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.1.1 Notification Constructor (INotification)

Initializes a new instance of the Notification [▶ 974] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Notification(
    INotification notification
)
```

Parameters

notification	Type: TwinCAT.Ads.INotification [▶ 969] The notification.

Reference

Notification Class [▶ 974]

Notification Overload [▶ 975]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.1.2 Notification Constructor (UInt32, DateTimeOffset, Object, ReadOnlyMemory`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Notification(
    uint handle,
    DateTimeOffset dateTime,
    Object userData,
    ReadOnlyMemory data
)
```

Parameters

handle	Type: System.UInt32
dateTime	Type: System.DateTimeOffset
userData	Type: System.Object
data	Type: ReadOnlyMemory

Reference

Notification Class [▶ 974]

Notification Overload [▶ 975]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.2 Notification Properties

The Notification [▶ 974] type exposes the following members.

Properties

	Name	Description
	Data [▶ 977]	The notification Data.
	Handle [▶ 977]	The notification handle
	TimeStamp [▶ 978]	Gets the time stamp of the INotification [▶ 969]
	UserData [▶ 978]	Attached UserData/Tag at the INotification [▶ 969]

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.2.1 Notification.Data Property

The notification Data.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemory Data { get; }
```

Property Value

Type: ReadOnlyMemory
The data.

Implements

INotification.Data [▶ 970]

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.2.2 Notification.Handle Property

The notification handle

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Handle { get; }
```

Property Value

Type: UInt32
The handle.

Implements

INotification.Handle [▶ 970]

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.2.3 Notification.TimeStamp Property

Gets the time stamp of the INotification [▶ 969]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The time stamp.

Implements

INotification.TimeStamp [▶ 971]

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.2.4 Notification.UserData Property

Attached UserData/Tag at the INotification [▶ 969]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object UserData { get; }
```

Property Value

Type: Object
The user data.

Implements

INotification.UserData [▶ 971]

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.52.3 Notification Methods

The Notification [▶ 974] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

Notification Class [▶ 974]

TwinCAT.Ads Namespace [▶ 151]

6.2.53 NotificationSettings Class

Notification communication settings

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.NotificationSettings
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class NotificationSettings : INotificationSettings,
    IComparable<INotificationSettings>
```

The NotificationSettings type exposes the following members.

Constructors

	Name	Description
	NotificationSettings (AdsTransMode, Int32, Int32) [▶ 982]	Initializes a new instance of the NotificationSettings class.
	NotificationSettings (AdsTransMode, TimeSpan, TimeSpan) [▶ 982]	Initializes a new instance of the NotificationSettings class.

Properties

	Name	Description
	CycleTime [▶ 983]	Gets or sets the cycle time (in milliseconds) for AdsNotifications.
	Default [▶ 984]	Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)
	ImmediatelyOnChange [▶ 984]	Gets the settings for a 'Immediate on change' notification.
	MaxDelay [▶ 985]	Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.
	NotificationMode [▶ 985]	Gets or sets the ADS Transmission mode.

Methods

	Name	Description
	CompareTo [▶ 986]	Compares this NotificationSettings in term of priorities to the other NotificationSettings.
	Equals [▶ 987]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode [▶ 987]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Operators

	Name	Description
	Equality [▶ 988]	Operator==
	Inequality [▶ 989]	Implements the != operator.

Remarks

AdsTransMode [▶ 639]	Parameter semantic
CyclicInContext [▶ 639]	Value of parameter is interpreted as task context number IAdsSymbol.ContextMask/>
OnChangeInContext [▶ 639]	Value of parameter is interpreted as task context number IAdsSymbol.ContextMask/>

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.53.1 NotificationSettings Constructor

Overload List

	Name	Description
	NotificationSettings (AdsTransMode, Int32, Int32) [▶ 982]	Initializes a new instance of the NotificationSettings [▶ 979] class.
	NotificationSettings (AdsTransMode, TimeSpan, TimeSpan) [▶ 982]	Initializes a new instance of the NotificationSettings [▶ 979] class.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.1.1 NotificationSettings Constructor (AdsTransMode, Int32, Int32)

Initializes a new instance of the NotificationSettings [▶ 979] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NotificationSettings(
    AdsTransMode mode,
    int cycleTime,
    int maxDelay
)
```

Parameters

mode	Type: TwinCAT.Ads.AdsTransMode [▶ 639] The ADS Transmission mode.
cycleTime	Type: System.Int32 The cycle time in ms.
maxDelay	Type: System.Int32 The maximum delay in ms

Reference

NotificationSettings Class [▶ 979]

NotificationSettings Overload [▶ 981]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.1.2 NotificationSettings Constructor (AdsTransMode, TimeSpan, TimeSpan)

Initializes a new instance of the NotificationSettings [▶ 979] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NotificationSettings(
    AdsTransMode mode,
    TimeSpan cycleTime,
    TimeSpan maxDelay
)
```

Parameters

mode	Type: TwinCAT.Ads.AdsTransMode [▶ 639] The ADS Transmission mode.
cycleTime	Type: System.TimeSpan The cycle time in ms.

maxDelay	Type: System.TimeSpan The maximum delay in ms

Reference

NotificationSettings Class [▶ 979]

NotificationSettings Overload [▶ 981]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2 NotificationSettings Properties

The NotificationSettings [▶ 979] type exposes the following members.

Properties

	Name	Description
	CycleTime [▶ 983]	Gets or sets the cycle time (in milliseconds) for AdsNotifications.
	Default [▶ 984]	Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)
	ImmediatelyOnChange [▶ 984]	Gets the settings for a 'Immediate on change' notification.
	MaxDelay [▶ 985]	Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.
	NotificationMode [▶ 985]	Gets or sets the ADS Transmission mode.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2.1 NotificationSettings.CycleTime Property

Gets or sets the cycle time (in milliseconds) for AdsNotifications.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int CycleTime { get; }
```

Property Value

Type: Int32
The cycle time.

Remarks

The ADS server checks if the value changes in this time slice. The unit is 1ms

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2.2 NotificationSettings.Default Property

Gets the default Settings (AdsTransMode.OnChange, CycleTime 200 ms, MaxDelay: off)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static NotificationSettings Default { get; }
```

Property Value

Type: NotificationSettings [▶ 979]
The default.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2.3 NotificationSettings.ImmediatelyOnChange Property

Gets the settings for a 'Immediate on change' notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static NotificationSettings ImmediatelyOnChange { get; }
```

Property Value

Type: NotificationSettings [▶ 979]
The immediately on change.

Remarks

AdsTransMode.OnChange, CycleTime: 0 ms, MaxDelay: off)

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2.4 NotificationSettings.MaxDelay Property

Gets or sets the Maximum Delay Time (in milliseconds) for AdsNotifications.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int MaxDelay { get; }
```

Property Value

Type: Int32
The maximum Delay time for ADS Notifications.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.2.5 NotificationSettings.NotificationMode Property

Gets or sets the ADS Transmission mode.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsTransMode NotificationMode { get; }
```

Property Value

Type: AdsTransMode [▶ 639]
The Transmission mode.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.3 NotificationSettings Methods

The NotificationSettings [▶ 979] type exposes the following members.

Methods

	Name	Description
	CompareTo [▶ 986]	Compares this NotificationSettings [▶ 979] in term of priorities to the other NotificationSettings [▶ 979].
	Equals [▶ 987]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 987]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.3.1 NotificationSettings.CompareTo Method

Compares this NotificationSettings [▶ 979] in term of priorities to the other NotificationSettings [▶ 979].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int CompareTo(
    INotificationSettings other
)
```

Parameters

other	Type: TwinCAT.Ads.INotificationSettings [▶ 972] The other.

Return Value

Type: Int32
1: this has higher priority (shorter times), 0: Equal, 1: Lower prioritiy

Implements

IComparable.T..CompareTo(T)

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

NotificationSettingsPriorityComparer

6.2.53.3.2 NotificationSettings.Equals Method

Equals

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.3.3 NotificationSettings.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.4 NotificationSettings Operators

The NotificationSettings [▶ 979] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 988]	Operator==
	Inequality [▶ 989]	Implements the != operator.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.4.1 NotificationSettings.Equality Operator

Operator==

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    NotificationSettings o1,
    NotificationSettings o2
)
```

Parameters

o1	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The o1.
o2	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.53.4.2 NotificationSettings.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    NotificationSettings o1,
    NotificationSettings o2
)
```

Parameters

o1	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The o1.
o2	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

NotificationSettings Class [▶ 979]

TwinCAT.Ads Namespace [▶ 151]

6.2.54 ResultAds Class

Base class for an (asynchronous) ADS Task Result

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds
    More... [▶ 991]
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultAds
```

The ResultAds type exposes the following members.

Constructors

	Name	Description
	ResultAds [▶ 991]	Initializes a new instance of the ResultAds class.

Properties

	Name	Description
	Empty [▶ 992]	Gets an empty ResultAds initialized to None [▶ 575].
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result object.
	Failed [▶ 993]	Gets a value indicating whether the ResultAds state is failed.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds state is succeeded.

Methods

	Name	Description
	CreateError(AdsErrorCode) [▶ 995]	Creates an Error Result.
	CreateError.TValue.(AdsErrorCode, TValue) [▶ 995]	Creates an Error Result.
	CreateSuccess. [▶ 996]	Creates a success result.
	CreateSuccess.TValue.(TValue) [▶ 997]	Creates a success result.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

The base class is used whenever an asynchronous task communicates via ADS and should return its AdsErrorCode [▶ 575] within its tasks result.

Reference

TwinCAT.Ads Namespace [▶ 151]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds
    TwinCAT.Ads.ResultAnyValue [▶ 998]
    TwinCAT.Ads.ResultDeviceInfo [▶ 1001]
    TwinCAT.Ads.ResultHandle [▶ 1005]
    TwinCAT.Ads.ResultRead [▶ 1008]
    TwinCAT.Ads.ResultReadAdsState [▶ 1010]
    TwinCAT.Ads.ResultReadDeviceState [▶ 1016]
    TwinCAT.Ads.ResultRpcMethod [▶ 1025]
    TwinCAT.Ads.ResultValue.TValue. [▶ 1029]
    TwinCAT.Ads.ResultWrite [▶ 1032]
    TwinCAT.Ads.SumCommand.ResultSumCommand [▶ 1210]

6.2.54.1 ResultAds Constructor

Initializes a new instance of the ResultAds [▶ 989] class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected ResultAds(
    AdsErrorCode errorCode
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The Ads ErrorCode

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.2 ResultAds Properties

The ResultAds [▶ 989] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 992]	Gets an empty ResultAds [▶ 989] initialized to None [▶ 575].

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object.
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.2.1 ResultAds.Empty Property

Gets an empty ResultAds [▶ 989] initialized to None [▶ 575].

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultAds Empty { get; }
```

Property Value

Type: ResultAds [▶ 989]
The empty.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.2.2 ResultAds.ErrorCode Property

Gets the ADS Error code bound to this Result [▶ 989] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode ErrorCode { get; }
```

Property Value

Type: AdsErrorCode [▶ 575]
The error code.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.2.3 ResultAds.Failed Property

Gets a value indicating whether the ResultAds [▶ 989] state is failed.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Failed { get; }
```

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.2.4 ResultAds.Succeeded Property

Gets a value indicating whether the ResultAds [▶ 989] state is succeeded.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Succeeded { get; }
```

Property Value

Type: Boolean
true if succeeded; otherwise, false.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.3 ResultAds Methods

The ResultAds [▶ 989] type exposes the following members.

Methods

	Name	Description
	CreateError(AdsErrorCode) [▶ 995]	Creates an Error Result.
	CreateError.TValue.(AdsErrorCode, TValue) [▶ 995]	Creates an Error Result.
	CreateSuccess. [▶ 996]	Creates a success result.
	CreateSuccess.TValue.(TValue) [▶ 997]	Creates a success result.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989]
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.3.1 ResultAds.CreateError Method

Overload List

	Name	Description
	CreateError(AdsErrorCode) [▶ 995]	Creates an Error Result.
	CreateError.TValue.(AdsErrorCode, TValue) [▶ 995]	Creates an Error Result.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

ResultAds.CreateError Method (AdsErrorCode)

Creates an Error Result.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultAds CreateError(
    AdsErrorCode errorCode
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.

Return Value

Type: ResultAds [▶ 989]
ResultValue<T>.

Reference

ResultAds Class [▶ 989]

CreateError Overload [▶ 994]

TwinCAT.Ads Namespace [▶ 151]

ResultAds.CreateError.TValue. Method (AdsErrorCode, TValue)

Creates an Error Result.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultValue<TValue> CreateError<TValue>(
    AdsErrorCode errorCode,
    TValue defaultValue
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
defaultValue	Type: TValue The default value.

Type Parameters

TValue	The type of the t value.

Return Value

Type: ResultValue [▶ 1029].TValue.
ResultValue<T>.

Reference

ResultAds Class [▶ 989]

CreateError Overload [▶ 994]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.3.2 ResultAds.CreateSuccess Method

Overload List

	Name	Description
	CreateSuccess. [▶ 996]	Creates a success result.
	CreateSuccess.TValue.(TValue) [▶ 997]	Creates a success result.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

ResultAds.CreateSuccess Method

Creates a success result.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultAds CreateSuccess()
```

Return Value

Type: ResultAds [▶ 989]
ResultValue<T>.

Reference

ResultAds Class [▶ 989]

CreateSuccess Overload [▶ 996]

TwinCAT.Ads Namespace [▶ 151]

ResultAds.CreateSuccess.TValue. Method (TValue)

Creates a success result.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultValue<TValue> CreateSuccess<TValue>(
    TValue value
)
```

Parameters

value	Type: TValue The value.

Type Parameters

TValue

Return Value

Type: ResultValue [▶ 1029].TValue.
ResultValue<T>.

Reference

ResultAds Class [▶ 989]

CreateSuccess Overload [▶ 996]

TwinCAT.Ads Namespace [▶ 151]

6.2.54.3.3 ResultAds.SetError Method

Sets the error state of this ResultAds [▶ 989]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void SetError(
    AdsErrorCode error
)
```

Parameters

error	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error.

Reference

ResultAds Class [▶ 989]

TwinCAT.Ads Namespace [▶ 151]

6.2.55 ResultAnyValue Class

Result object for asynchronous reading an 'AnyValue'/Primitive Value via tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultAnyValue

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultAnyValue : ResultAds
```

The ResultAnyValue type exposes the following members.

Constructors

	Name	Description
	ResultAnyValue [▶ 999]	Initializes a new instance of the ResultAnyValue struct.

Properties

	Name	Description
	Empty [▶ 1000]	Gets the empty ResultAnyValue object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1000]	Gets the read value.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.55.1 ResultAnyValue Constructor

Initializes a new instance of the ResultAnyValue [▶ 998] struct.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultAnyValue(
    AdsErrorCode errorCode,
    Object value
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
value	Type: System.Object The value.

Reference

ResultAnyValue Class [▶ 998]

TwinCAT.Ads Namespace [▶ 151]

6.2.55.2 ResultAnyValue Properties

The ResultAnyValue [▶ 998] type exposes the following members.

Properties

	Name	Description
 	Empty [▶ 1000]	Gets the empty ResultAnyValue [▶ 998] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1000]	Gets the read value.

Reference

ResultAnyValue Class [▶ 998]

TwinCAT.Ads Namespace [▶ 151]

6.2.55.2.1 ResultAnyValue.Empty Property

Gets the empty ResultAnyValue [▶ 998] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultAnyValue Empty { get; }
```

Property Value

Type: ResultAnyValue [▶ 998]
The empty / unprocessed result.

Reference

ResultAnyValue Class [▶ 998]

TwinCAT.Ads Namespace [▶ 151]

6.2.55.2.2 ResultAnyValue.Value Property

Gets the read value.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object Value { get; }
```

Property Value

Type: Object

Reference

ResultAnyValue Class [▶ 998]

TwinCAT.Ads Namespace [▶ 151]

6.2.55.3 ResultAnyValue Methods

The ResultAnyValue [▶ 998] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultAnyValue Class [▶ 998]

TwinCAT.Ads Namespace [▶ 151]

6.2.56 ResultDeviceInfo Class

Ads Task Result for DeviceInfo [▶ 1003] requests (async operations).

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultDeviceInfo

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultDeviceInfo : ResultAds
```

The ResultDeviceInfo type exposes the following members.

Properties

	Name	Description
	DeviceInfo [▶ 1003]	Gets the device information (Task result)
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Name [▶ 1003]	The name of the Device
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Version [▶ 1004]	The ADS Version of the the Device.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.56.1 ResultDeviceInfo Properties

The ResultDeviceInfo [▶ 1001] type exposes the following members.

Properties

	Name	Description
	DeviceInfo [▶ 1003]	Gets the device information (Task result)

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Name [▶ 1003]	The name of the Device
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Version [▶ 1004]	The ADS Version of the the Device.

Reference

ResultDeviceInfo Class [▶ 1001]

TwinCAT.Ads Namespace [▶ 151]

6.2.56.1.1 ResultDeviceInfo.DeviceInfo Property

Gets the device information (Task result)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DeviceInfo DeviceInfo { get; }
```

Property Value

Type: DeviceInfo [▶ 698]
The device information.

Reference

ResultDeviceInfo Class [▶ 1001]

TwinCAT.Ads Namespace [▶ 151]

6.2.56.1.2 ResultDeviceInfo.Name Property

The name of the Device

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String

Reference

ResultDeviceInfo Class [▶ 1001]

TwinCAT.Ads Namespace [▶ 151]

6.2.56.1.3 ResultDeviceInfo.Version Property

The ADS Version of the the Device.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsVersion Version { get; }
```

Property Value

Type: AdsVersion [▶ 642]

Reference

ResultDeviceInfo Class [▶ 1001]

TwinCAT.Ads Namespace [▶ 151]

6.2.56.2 ResultDeviceInfo Methods

The ResultDeviceInfo [▶ 1001] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultDeviceInfo Class [▶ 1001]

TwinCAT.Ads Namespace [▶ 151]

6.2.57 ResultHandle Class

Result object for asynchronous registering an ADS Handle via tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultHandle

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultHandle : ResultAds
```

The ResultHandle type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1006]	Gets the empty ResultHandle object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handle [▶ 1007]	Gets the registered handle
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.57.1 ResultHandle Properties

The ResultHandle [▶ 1005] type exposes the following members.

Properties

	Name	Description
S	Empty [▶ 1006]	Gets the empty ResultHandle [▶ 1005] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handle [▶ 1007]	Gets the registered handle
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultHandle Class [▶ 1005]

TwinCAT.Ads Namespace [▶ 151]

6.2.57.1.1 ResultHandle.Empty Property

Gets the empty ResultHandle [▶ 1005] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultHandle Empty { get; }
```

Property Value

Type: ResultHandle [▶ 1005]
The empty / unprocessed result.

Reference

ResultHandle Class [▶ 1005]

TwinCAT.Ads Namespace [▶ 151]

6.2.57.1.2 ResultHandle.Handle Property

Gets the registered handle

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Handle { get; }
```

Property Value

Type: UInt32

Reference

ResultHandle Class [▶ 1005]

TwinCAT.Ads Namespace [▶ 151]

6.2.57.2 ResultHandle Methods

The ResultHandle [▶ 1005] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultHandle Class [▶ 1005]

TwinCAT.Ads Namespace [▶ 151]

6.2.58 ResultRead Class

Asynchronous ADS Read result.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultRead
      TwinCAT.Ads.ResultReadBytes [▶ 1013]
      TwinCAT.Ads.ResultReadWrite [▶ 1019]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultRead : ResultAds
```

The ResultRead type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1009]	Gets the empty ResultRead object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.58.1 ResultRead Properties

The ResultRead [▶ 1008] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1009]	Gets the empty ResultRead [▶ 1008] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultRead Class [▶ 1008]

TwinCAT.Ads Namespace [▶ 151]

6.2.58.1.1 ResultRead.Empty Property

Gets the empty ResultRead [▶ 1008] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultRead Empty { get; }
```

Property Value

Type: ResultRead [▶ 1008]
The empty / unprocessed result.

Reference

ResultRead Class [▶ 1008]

TwinCAT.Ads Namespace [▶ 151]

6.2.58.1.2 ResultRead.ReadBytes Property

Gets the number of Read bytes.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReadBytes { get; }
```

Property Value

Type: Int32

Reference

ResultRead Class [▶ 1008]

TwinCAT.Ads Namespace [▶ 151]

6.2.58.2 ResultRead Methods

The ResultRead [▶ 1008] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultRead Class [▶ 1008]

TwinCAT.Ads Namespace [▶ 151]

6.2.59 ResultReadAdsState Class

Result object for asynchronous reading AdsStates via tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultReadAdsState

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadAdsState : ResultAds
```

The ResultReadAdsState type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1012]	Gets the empty ResultReadAdsState object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	State [▶ 1012]	Gets the ADS state.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.59.1 ResultReadAdsState Properties

The ResultReadAdsState [▶ 1010] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1012]	Gets the empty ResultReadAdsState [▶ 1010] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	State [▶ 1012]	Gets the ADS state.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultReadAdsState Class [▶ 1010]

TwinCAT.Ads Namespace [▶ 151]

6.2.59.1.1 ResultReadAdsState.Empty Property

Gets the empty ResultReadAdsState [▶ 1010] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadAdsState Empty { get; }
```

Property Value

Type: ResultReadAdsState [▶ 1010]
The empty / unprocessed result.

Reference

ResultReadAdsState Class [▶ 1010]

TwinCAT.Ads Namespace [▶ 151]

6.2.59.1.2 ResultReadAdsState.State Property

Gets the ADS state.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsState State { get; }
```

Property Value

Type: AdsState [▶ 626]

Reference

ResultReadAdsState Class [▶ 1010]

TwinCAT.Ads Namespace [▶ 151]

6.2.59.2 ResultReadAdsState Methods

The ResultReadAdsState [▶ 1010] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadAdsState Class [▶ 1010]

TwinCAT.Ads Namespace [▶ 151]

6.2.60 ResultReadBytes Class

ADS Task Result returning Read data for async Read operations.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultRead [▶ 1008]
      TwinCAT.Ads.ResultReadBytes
        TwinCAT.Ads.ResultReadWriteBytes [▶ 1022]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadBytes : ResultRead
```

The ResultReadBytes type exposes the following members.

Properties

	Name	Description
	Data [▶ 1015]	The read bytes as continuous region of memory.
	Empty [▶ 1016]	Gets the empty ResultReadBytes object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.60.1 ResultReadBytes Properties

The ResultReadBytes [▶ 1013] type exposes the following members.

Properties

	Name	Description
	Data [▶ 1015]	The read bytes as continuous region of memory.
S	Empty [▶ 1016]	Gets the empty ResultReadBytes [▶ 1013] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultReadBytes Class [▶ 1013]

TwinCAT.Ads Namespace [▶ 151]

6.2.60.1.1 ResultReadBytes.Data Property

The read bytes as continuous region of memory.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemory Data { get; }
```

Property Value

Type: ReadOnlyMemory

Reference

ResultReadBytes Class [▶ 1013]

TwinCAT.Ads Namespace [▶ 151]

6.2.60.1.2 ResultReadBytes.Empty Property

Gets the empty ResultReadBytes [▶ 1013] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadBytes Empty { get; }
```

Property Value

Type: ResultReadBytes [▶ 1013]
The empty / unprocessed result.

Reference

ResultReadBytes Class [▶ 1013]

TwinCAT.Ads Namespace [▶ 151]

6.2.60.2 ResultReadBytes Methods

The ResultReadBytes [▶ 1013] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadBytes Class [▶ 1013]

TwinCAT.Ads Namespace [▶ 151]

6.2.61 ResultReadDeviceState Class

Result object for asynchronous ADS ReadDeviceState tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultReadDeviceState

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadDeviceState : ResultAds
```

The ResultReadDeviceState type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1018]	Gets the empty ResultReadDeviceState object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	State [▶ 1018]	The Device state.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.61.1 ResultReadDeviceState Properties

The ResultReadDeviceState [▶ 1016] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1018]	Gets the empty ResultReadDeviceState [▶ 1016] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	State [▶ 1018]	The Device state.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultReadDeviceState Class [▶ 1016]

TwinCAT.Ads Namespace [▶ 151]

6.2.61.1.1 ResultReadDeviceState.Empty Property

Gets the empty ResultReadDeviceState [▶ 1016] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadDeviceState Empty { get; }
```

Property Value

Type: ResultReadDeviceState [▶ 1016]
The empty / unprocessed result.

Reference

ResultReadDeviceState Class [▶ 1016]

TwinCAT.Ads Namespace [▶ 151]

6.2.61.1.2 ResultReadDeviceState.State Property

The Device state.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo State { get; }
```

Property Value

Type: StateInfo [▶ 1041]

Reference

ResultReadDeviceState Class [▶ 1016]

TwinCAT.Ads Namespace [▶ 151]

6.2.61.2 ResultReadDeviceState Methods

The ResultReadDeviceState [▶ 1016] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadDeviceState Class [▶ 1016]

TwinCAT.Ads Namespace [▶ 151]

6.2.62 ResultReadWrite Class

Result object for asynchronous ADS ReadWrite tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultRead [▶ 1008]
      TwinCAT.Ads.ResultReadWrite

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadWrite : ResultRead
```

The ResultReadWrite type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1021]	Gets the empty ResultReadWrite object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.62.1 ResultReadWrite Properties

The ResultReadWrite [▶ 1019] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1021]	Gets the empty ResultReadWrite [▶ 1019] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultReadWrite Class [▶ 1019]

TwinCAT.Ads Namespace [▶ 151]

6.2.62.1.1 ResultReadWrite.Empty Property

Gets the empty ResultReadWrite [▶ 1019] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadWrite Empty { get; }
```

Property Value

Type: ResultReadWrite [▶ 1019]
The empty / unprocessed result.

Reference

ResultReadWrite Class [▶ 1019]

TwinCAT.Ads Namespace [▶ 151]

6.2.62.2 ResultReadWrite Methods

The ResultReadWrite [▶ 1019] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadWrite Class [▶ 1019]

TwinCAT.Ads Namespace [▶ 151]

6.2.63 ResultReadWriteBytes Class

Result object for asynchronous ADS ReadWrite tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultRead [▶ 1008]
      TwinCAT.Ads.ResultReadBytes [▶ 1013]
        TwinCAT.Ads.ResultReadWriteBytes

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadWriteBytes : ResultReadBytes
```

The ResultReadWriteBytes type exposes the following members.

Properties

	Name	Description
	Data [▶ 1015]	The read bytes as continuous region of memory. (Inherited from ResultReadBytes [▶ 1013].)
S	Empty [▶ 1024]	Gets the empty ResultReadWriteBytes object.

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.63.1 ResultReadWriteBytes Properties

The ResultReadWriteBytes [▶ 1022] type exposes the following members.

Properties

	Name	Description
	Data [▶ 1015]	The read bytes as continuous region of memory. (Inherited from ResultReadBytes [▶ 1013].)
S	Empty [▶ 1024]	Gets the empty ResultReadWriteBytes [▶ 1022] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBytes [▶ 1010]	Gets the number of Read bytes. (Inherited from ResultRead [▶ 1008].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultReadWriteBytes Class [▶ 1022]

TwinCAT.Ads Namespace [▶ 151]

6.2.63.1.1 ResultReadWriteBytes.Empty Property

Gets the empty ResultReadWriteBytes [▶ 1022] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadWriteBytes Empty { get; }
```

Property Value

Type: ResultReadWriteBytes [▶ 1022]
The empty / unprocessed result.

Reference

ResultReadWriteBytes Class [▶ 1022]

TwinCAT.Ads Namespace [▶ 151]

6.2.63.2 ResultReadWriteBytes Methods

The ResultReadWriteBytes [▶ 1022] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadWriteBytes Class [▶ 1022]

TwinCAT.Ads Namespace [▶ 151]

6.2.64 ResultRpcMethod Class

Class representing a result of an asynchronous RpcMethod call. Implements the ResultAds [▶ 989]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultRpcMethod

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultRpcMethod : ResultAds
```

The ResultRpcMethod type exposes the following members.

Constructors

	Name	Description
	ResultRpcMethod [▶ 1026]	Initializes a new instance of the ResultRpcMethod struct.

Properties

	Name	Description
	Empty [▶ 1027]	Gets the Empty Result (initialized to None [▶ 575] and default ReturnValue [▶ 1028] (NULL).
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	OutValues [▶ 1027]	Gets the output parameter values.
	ReturnValue [▶ 1028]	The (optional) return value of the RPC Method.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

Optionally this Result holds an ReturnValue [▶ 1028] if the RpcCall is not void.

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.ResultAds [▶ 989]

6.2.64.1 ResultRpcMethod Constructor

Initializes a new instance of the ResultRpcMethod [▶ 1025] struct.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultRpcMethod(
    AdsErrorCode errorCode,
    Object returnValue,
    Object[] outParameters
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
returnValue	Type: System.Object The value.
outParameters	Type: .System.Object. The out parameters.

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.64.2 ResultRpcMethod Properties

The ResultRpcMethod [▶ 1025] type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1027]	Gets the Empty Result (initialized to None [▶ 575] and default ReturnValue [▶ 1028] (NULL).
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	OutValues [▶ 1027]	Gets the output parameter values.
	ReturnValue [▶ 1028]	The (optional) return value of the RPC Method.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.64.2.1 ResultRpcMethod.Empty Property

Gets the Empty Result (initialized to None [▶ 575] and default ReturnValue [▶ 1028] (NULL).

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultRpcMethod Empty { get; }
```

Property Value

Type: ResultRpcMethod [▶ 1025]
The empty.

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.64.2.2 ResultRpcMethod.OutValues Property

Gets the output parameter values.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object[] OutValues { get; }
```

Property Value

Type: .Object.
The output parameters.

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.64.2.3 ResultRpcMethod.ReturnValue Property

The (optional) return value of the RPC Method.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReturnValue { get; }
```

Property Value

Type: Object

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.64.3 ResultRpcMethod Methods

The ResultRpcMethod [▶ 1025] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultRpcMethod Class [▶ 1025]

TwinCAT.Ads Namespace [▶ 151]

6.2.65 ResultValue.TValue. Class

ADS Result object returning a generic value result (TValue) (asynchronous read). Implements the ResultAds [▶ 989]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultValue.TValue.
      TwinCAT.TypeSystem.ResultDataTypes [▶ 2347]
      TwinCAT.TypeSystem.ResultSymbols.T. [▶ 2356]

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultValue<TValue> : ResultAds
```

Type Parameters

TValue	The type of the result value.

The ResultValue.TValue. type exposes the following members.

Constructors

	Name	Description
	ResultValue.TValue. [▶ 1030]	Initializes a new instance of the ResultValue class.

Properties

	Name	Description
S	Empty [▶ 1031]	Gets the Empty result initialized to NoError [▶ 575] and the default of TValue.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)

	Name	Description
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1032]	The value object.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

The ResultValue.TValue. type is used in generic ADS read operations, where a value is returned.

Reference

TwinCAT.Ads Namespace [▶ 151]

TwinCAT.Ads.ResultAds [▶ 989]

6.2.65.1 ResultValue.TValue. Constructor

Initializes a new instance of the ResultValue class.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultValue(
    AdsErrorCode errorCode,
    TValue value
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
value	Type: TValue [▶ 1029] The value.

Reference

ResultValue.TValue. Class [▶ 1029]

TwinCAT.Ads Namespace [▶ 151]

6.2.65.2 ResultValue.TValue. Properties

The ResultValue.TValue. [▶ 1029] generic type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1031]	Gets the Empty result initialized to NoError [▶ 575] and the default of TValue.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1032]	The value object.

Reference

ResultValue.TValue. Class [▶ 1029]

TwinCAT.Ads Namespace [▶ 151]

6.2.65.2.1 ResultValue.TValue..Empty Property

Gets the Empty result initialized to NoError [▶ 575] and the default of TValue.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultValue<TValue> Empty { get; }
```

Property Value

Type: ResultValue [▶ 1029].TValue [▶ 1029].
The empty.

Reference

ResultValue.TValue. Class [▶ 1029]

TwinCAT.Ads Namespace [▶ 151]

6.2.65.2.2 ResultValue.TValue..Value Property

The value object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TValue Value { get; }
```

Property Value

Type: TValue [▶ 1029]

Reference

ResultValue.TValue. Class [▶ 1029]

TwinCAT.Ads Namespace [▶ 151]

6.2.65.3 ResultValue.TValue. Methods

The ResultValue.TValue. [▶ 1029] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultValue.TValue. Class [▶ 1029]

TwinCAT.Ads Namespace [▶ 151]

6.2.66 ResultWrite Class

Result for asynchronous ADS write tasks.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultWrite

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultWrite : ResultAds
```

The ResultWrite type exposes the following members.

Properties

	Name	Description
	Empty [▶ 1034]	Gets the empty ResultWrite object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.66.1 ResultWrite Properties

The ResultWrite [▶ 1032] type exposes the following members.

Properties

	Name	Description
 	Empty [▶ 1034]	Gets the empty ResultWrite [▶ 1032] object.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultWrite Class [▶ 1032]

TwinCAT.Ads Namespace [▶ 151]

6.2.66.1.1 ResultWrite.Empty Property

Gets the empty ResultWrite [▶ 1032] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultWrite Empty { get; }
```

Property Value

Type: ResultWrite [▶ 1032]
The empty / unprocessed result.

Reference

ResultWrite Class [▶ 1032]

TwinCAT.Ads Namespace [▶ 151]

6.2.66.2 ResultWrite Methods

The ResultWrite [▶ 1032] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultWrite Class [▶ 1032]

TwinCAT.Ads Namespace [▶ 151]

6.2.67 SessionSettings Class

Session settings class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.SessionSettings
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SessionSettings : IAdsSessionSettings
```

The SessionSettings type exposes the following members.

Properties

	Name	Description
	Default [▶ 1037]	Gets the default Settings (Synchronized).
	DefaultCommunicationTimeout [▶ 1037]	The default communication timeout (5 Seconds)
	DefaultResurrectionTime [▶ 1038]	The default resurrection time (21 Seconds)
	FastWriteThrough [▶ 1038]	Gets a Settings object that configures the AdsSession for FastWriteThrough
	ResurrectionTime [▶ 1039]	Gets or sets the resurrection time (Default: DefaultResurrectionTime [▶ 1038])

	Name	Description
	SymbolLoader [▶ 1039]	Gets or sets the symbol loader settings
	Timeout [▶ 1040]	Gets the ADS timeout in milliseconds.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1 SessionSettings Properties

The SessionSettings [▶ 1035] type exposes the following members.

Properties

	Name	Description
S	Default [▶ 1037]	Gets the default Settings (Synchronized).
S	DefaultCommunicationTimeout [▶ 1037]	The default communication timeout (5 Seconds)
S	DefaultResurrectionTime [▶ 1038]	The default resurrection time (21 Seconds)
S	FastWriteThrough [▶ 1038]	Gets a Settings object that configures the AdsSession for FastWriteThrough
	ResurrectionTime [▶ 1039]	Gets or sets the resurrection time (Default: DefaultResurrectionTime [▶ 1038])
	SymbolLoader [▶ 1039]	Gets or sets the symbol loader settings
	Timeout [▶ 1040]	Gets the ADS timeout in milliseconds.

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.1 SessionSettings.Default Property

Gets the default Settings (Synchronized).

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SessionSettings Default { get; }
```

Property Value

Type: SessionSettings [▶ 1035]
The default settings.

Remarks

The following defaults are set here:

Setting	Description
Communication Timeout (Timeout [▶ 1040])	Default communication timeout (DefaultCommunicationTimeout [▶ 1037], default 5s)
Resurrection Timeout (ResurrectionTime [▶ 1039])	Default communication timeout (DefaultResurrectionTime [▶ 1038], default 21s)
Dynamic SymbolLoader settings SymbolLoader [▶ 1039]	Synchronized mode activated (DefaultDynamic [▶ 146])

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.2 SessionSettings.DefaultCommunicationTimeout Property

The default communication timeout (5 Seconds)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan DefaultCommunicationTimeout { get; }
```

Property Value

Type: TimeSpan

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.3 SessionSettings.DefaultResurrectionTime Property

The default resurrection time (21 Seconds)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan DefaultResurrectionTime { get; }
```

Property Value

Type: TimeSpan

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.4 SessionSettings.FastWriteThrough Property

Gets a Settings object that configures the AdsSession for FastWriteThrough

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SessionSettings FastWriteThrough { get; }
```

Property Value

Type: SessionSettings [▶ 1035]
Session settings for a fast write through (with 200 ms Timeout).

Remarks

The settings typically can be used for polling clients, where the "FailFast" feature will be bypassed. That means, that communication fails doesn't trigger the FailFast interceptor and every Request will go out via ADS. This has the Drawback that communication Timeouts are longer and subsequent timeouts block the ADS mailbox (with the danger of overflows). So use this setting with care for specific purposes and should not be used for standard communication.

- No Resurrection time and therefore:
- No FailFastHandler active.
- Default communication timeout 200ms.
- Not synchronized Notifications.

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.5 SessionSettings.ResurrectionTime Property

Gets or sets the resurrection time (Default: DefaultResurrectionTime [▶ 1038])

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TimeSpan ResurrectionTime { get; set; }
```

Property Value

Type: TimeSpan
The resurrection time.

Implements

IAdsSessionSettings.ResurrectionTime [▶ 913]

Remarks

The resurrection time is the time after a lost connection Lost [▶ 67] can be 'resurrected'. This time is set to 21 Seconds by default (a value greater than the standard Ethernet connection timeout of 20s). The reason for this timeout is not to flood the ADS mailbox with requests that cannot be handled by the ethernet infrastructure. As long this Timespan is not expired after a recognized Lost [▶ 67], no further data communication is done, and requests are immediately ('FailFast') answered by communication exceptions. **Change this value only for edge cases.**

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.6 SessionSettings.SymbolLoader Property

Gets or sets the symbol loader settings

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolLoaderSettings SymbolLoader { get; set; }
```

Property Value

Type: SymbolLoaderSettings [▶ 140]
The symbol loader.

Implements

IAdsSessionSettings.SymbolLoader [▶ 914]

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.1.7 SessionSettings.Timeout Property

Gets the ADS timeout in milliseconds.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Timeout { get; }
```

Property Value

Type: Int32
The timeout.

Implements

IAdsSessionSettings.Timeout [▶ 914]

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.67.2 SessionSettings Methods

The SessionSettings [▶ 1035] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SessionSettings Class [▶ 1035]

TwinCAT.Ads Namespace [▶ 151]

6.2.68 StateInfo Structure

The structure contains the ADS state and device state.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public struct StateInfo : IEquatable<StateInfo>
```

The StateInfo type exposes the following members.

Constructors

	Name	Description
	StateInfo(ReadOnlySpan) [▶ 1042]	
	StateInfo(AdsState, Int16) [▶ 1043]	Initializes a new Instance of the StateInfo struct.

Properties

	Name	Description
	AdsState [▶ 1044]	Gets or sets the ADS state of this StateInfo object.
	DeviceState [▶ 1044]	Gets or sets the device state of this StateInfo object.
	Empty [▶ 1045]	Empty / Invalid / Uninitialized state.

Methods

	Name	Description
	Equals(Object) [▶ 1046]	Determines whether the specified Object is equal to this instance. (Overrides ValueType.Equals(Object).)
	Equals(StateInfo) [▶ 1046]	Determines whether the specified StateInfo is equal to this instance.
	GetHashCode [▶ 1047]	Returns a hash code for this instance. (Overrides ValueType.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType.)

Operators

	Name	Description
	Equality [▶ 1048]	Implements the ==.
	Inequality [▶ 1048]	Implements the !=.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.68.1 StateInfo Constructor

Overload List

	Name	Description
	StateInfo(ReadOnlySpan) [▶ 1042]	
	StateInfo(AdsState, Int16) [▶ 1043]	Initializes a new Instance of the StateInfo struct.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.1.1 StateInfo Constructor (ReadOnlySpan`1)

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo(
    ReadOnlySpan data
)
```

Parameters

data	Type: ReadOnlySpan

Reference

StateInfo Structure [▶ 1041]

StateInfo Overload [▶ 1042]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.1.2 StateInfo Constructor (AdsState, Int16)

Initializes a new Instance of the StateInfo struct.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StateInfo(
    AdsState adsState,
    short deviceState
)
```

Parameters

adsState	Type: TwinCAT.Ads.AdsState [▶ 626] Ads state.
deviceState	Type: System.Int16 Device state.

Reference

StateInfo Structure [▶ 1041]

StateInfo Overload [▶ 1042]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.2 StateInfo Properties

The StateInfo [▶ 1041] type exposes the following members.

Properties

	Name	Description
	AdsState [▶ 1044]	Gets or sets the ADS state of this StateInfo [▶ 1041] object.
	DeviceState [▶ 1044]	Gets or sets the device state of this StateInfo [▶ 1041] object.
S	Empty [▶ 1045]	Empty / Invalid / Uninitialized state.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.2.1 StateInfo.AdsState Property

Gets or sets the ADS state of this StateInfo [▶ 1041] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsState AdsState { get; set; }
```

Property Value

Type: AdsState [▶ 626]

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.2.2 StateInfo.DeviceState Property

Gets or sets the device state of this StateInfo [▶ 1041] object.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public short DeviceState { get; set; }
```

Property Value

Type: Int16

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.2.3 StateInfo.Empty Property

Empty / Invalid / Uninitialized state.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static StateInfo Empty { get; }
```

Property Value

Type: StateInfo [▶ 1041]
The empty.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.3 StateInfo Methods

The StateInfo [▶ 1041] type exposes the following members.

Methods

	Name	Description
	Equals(Object) [▶ 1046]	Determines whether the specified Object is equal to this instance. (Overrides ValueType.Equals(Object).)
	Equals(StateInfo) [▶ 1046]	Determines whether the specified StateInfo [▶ 1041] is equal to this instance.
	GetHashCode [▶ 1047]	Returns a hash code for this instance. (Overrides ValueType.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns the fully qualified type name of this instance. (Inherited from ValueType.)

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.3.1 StateInfo.Equals Method

Overload List

	Name	Description
	Equals(Object) [▶ 1046]	Determines whether the specified Object is equal to this instance. (Overrides ValueType.Equals(Object).)
	Equals(StateInfo) [▶ 1046]	Determines whether the specified StateInfo [▶ 1041] is equal to this instance.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

StateInfo.Equals Method (Object)

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object ob
)
```

Parameters

ob	Type: System.Object The Object to compare with this instance.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

StateInfo Structure [▶ 1041]

Equals Overload [▶ 1046]

TwinCAT.Ads Namespace [▶ 151]

StateInfo.Equals Method (StateInfo)

Determines whether the specified StateInfo [▶ 1041] is equal to this instance.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Equals(
    StateInfo info
)
```

Parameters

info	Type: TwinCAT.Ads.StateInfo [▶ 1041] The StateInfo [▶ 1041] to compare with this instance.

Return Value

Type: Boolean
true if the specified StateInfo [▶ 1041] is equal to this instance; otherwise, false.

Implements

IEquatable.T..Equals(T)

Reference

StateInfo Structure [▶ 1041]

Equals Overload [▶ 1046]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.3.2 StateInfo.GetHashCode Method

Returns a hash code for this instance.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.4 StateInfo Operators

The StateInfo [▶ 1041] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 1048]	Implements the ==.
	Inequality [▶ 1048]	Implements the !=.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.4.1 StateInfo.Equality Operator

Implements the ==.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    StateInfo a,
    StateInfo b
)
```

Parameters

a	Type: TwinCAT.Ads.StateInfo [▶ 1041] a.
b	Type: TwinCAT.Ads.StateInfo [▶ 1041] The b.

Return Value

Type: Boolean
The result of the operator.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.68.4.2 StateInfo.Inequality Operator

Implements the !=.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    StateInfo a,
    StateInfo b
)
```

Parameters

a	Type: TwinCAT.Ads.StateInfo [▶ 1041] a.
b	Type: TwinCAT.Ads.StateInfo [▶ 1041] The b.

Return Value

Type: Boolean
The result of the operator.

Reference

StateInfo Structure [▶ 1041]

TwinCAT.Ads Namespace [▶ 151]

6.2.69 TaskExtensions Class

TaskExtensions for Task Cancellation and Timeout

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TaskExtensions
Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class TaskExtensions
```

The TaskExtensions type exposes the following members.

Methods

	Name	Description
	WithCancellation.T. [▶ 1050]	Extends a worker Task with cancellation.
	WithCancellationAndTimeout [▶ 1051]	Extends a worker task with timeout and Cancellation
	WithTimeout [▶ 1052]	Extends a worker task with a timeout.

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.69.1 TaskExtensions Methods

The TaskExtensions [▶ 1049] type exposes the following members.

Methods

	Name	Description
	WithCancellation.T. [▶ 1050]	Extends a worker Task with cancellation.
	WithCancellationAndTimeout [▶ 1051]	Extends a worker task with timeout and Cancellation
	WithTimeout [▶ 1052]	Extends a worker task with a timeout.

Reference

TaskExtensions Class [▶ 1049]

TwinCAT.Ads Namespace [▶ 151]

6.2.69.1.1 TaskExtensions.WithCancellation.T. Method

Extends a worker Task with cancellation.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static Task<T> WithCancellation<T>(
    this Task<T> worker,
    CancellationToken cancellationToken
)
```

Parameters

worker	Type: System.Threading.Tasks.Task.T. The worker task.
cancellationToken	Type: System.Threading.CancellationToken The cancellation token.

Type Parameters

T	The Task Result

Return Value

Type: Task.T.
Task<T>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type Task.T.. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Exceptions

Exception	Condition
OperationCanceledException	

Reference

TaskExtensions Class [▶ 1049]

TwinCAT.Ads Namespace [▶ 151]

6.2.69.1.2 TaskExtensions.WithCancellationAndTimeout Method

Extends a worker task with timeout and Cancellation

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static Task<AdsErrorCode> WithCancellationAndTimeout(
    this Task<AdsErrorCode> worker,
    TimeSpan timeout,
    CancellationToken cancellationToken
)
```

Parameters

worker	Type: System.Threading.Tasks.Task.AdsErrorCode [▶ 575]. The worker task.
timeout	Type: System.TimeSpan The timeout.
cancellationToken	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
Task<AdsErrorCode>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type Task.AdsErrorCode [▶ 575].. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Exceptions

Exception	Condition
OperationCanceledException	

Reference

TaskExtensions Class [▶ 1049]

TwinCAT.Ads Namespace [▶ 151]

6.2.69.1.3 TaskExtensions.WithTimeout Method

Extends a worker task with a timeout.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static Task<AdsErrorCode> WithTimeout(
    this Task<AdsErrorCode> worker,
    TimeSpan timeout
)
```

Parameters

worker	Type: System.Threading.Tasks.Task.AdsErrorCode [▶ 575]. The worker task
timeout	Type: System.TimeSpan The timeout.

Return Value

Type: Task.AdsErrorCode [▶ 575].
Task<AdsErrorCode>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type Task.AdsErrorCode [▶ 575].. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

TaskExtensions Class [▶ 1049]

TwinCAT.Ads Namespace [▶ 151]

6.2.70 TransportProtocols Enumeration

Enum ADS TransportProtocol

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum TransportProtocols
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Router	1	ADS via Router
	TcpIp	2	ADS via TCP/IP (without router)
	All	3	Indicates that Router and TcpIp are appropriate (for establishing connections)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.71 ValueNotificationEventArgs.T. Class

Arguments for AdsNotificationEx [▶ 868] events.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.AdsNotificationEventArgs [▶ 597]
      TwinCAT.Ads.ValueNotificationEventArgs.T.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ValueNotificationEventArgs<T> : AdsNotificationEventArgs
```

Type Parameters

T

The ValueNotificationEventArgs.T. type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Handle [▶ 599]	Gets the Notification handle. (Inherited from AdsNotificationEventArgs [▶ 597].)
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [▶ 597].)
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Value [▶ 1055]	Value of the ADS Notification.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads Namespace [▶ 151]

6.2.71.1 ValueNotificationEventArgs.T. Properties

The ValueNotificationEventArgs.T. [▶ 1053] generic type exposes the following members.

Properties

	Name	Description
	Data [▶ 598]	Memory object holding the Notification Data/Value. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Handle [▶ 599]	Gets the Notification handle. (Inherited from AdsNotificationEventArgs [▶ 597].)
	TimeStamp [▶ 600]	Gets the time stamp of this Notification as DateTimeOffset. (Inherited from AdsNotificationEventArgs [▶ 597].)
	UserData [▶ 600]	Gets the user object. This object is passed by to AddDeviceNotification and can be used to store data. (Inherited from AdsNotificationEventArgs [▶ 597].)
	Value [▶ 1055]	Value of the ADS Notification.

Reference

ValueNotificationEventArgs.T. Class [▶ 1053]

TwinCAT.Ads Namespace [▶ 151]

6.2.71.1.1 ValueNotificationEventArgs.T..Value Property

Value of the ADS Notification.

Namespace: TwinCAT.Ads [▶ 151]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Value { get; }
```

Property Value

Type: T [▶ 1053]

Reference

ValueNotificationEventArgs.T. Class [▶ 1053]

TwinCAT.Ads Namespace [▶ 151]

6.2.71.2 ValueNotificationEventArgs.T. Methods

The ValueNotificationEventArgs.T. [▶ 1053] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ValueNotificationEventArgs.T. Class [▶ 1053]

TwinCAT.Ads Namespace [▶ 151]

6.3 TwinCAT.Ads.Reactive Namespace

Reactive Extensions for the ADS Client. All types within are contained in the ADS companion package "Beckhoff.TwinCAT.Ads.Reactive" which must be referenced seperately. https://www.nuget.org/packages/Beckhoff.TwinCAT.Ads.Reactive/

Classes

	Class	Description
	AdsClientExtensions [▶ 1056]	Extension class for AdsClient respective IAdsConnection [▶ 765] to provide reactive ADS extensions.
	AnyTypeExtensions [▶ 1075]	Extension class for IAdsConnection [▶ 765] to provide reactive ADS extensions (accessing symbol value sequences with the ANY_TYPE concept)
	SymbolValueNotification [▶ 1104]	Symbol Notification class
	ValueSymbolExtensions [▶ 1106]	Extension class for IAdsConnection [▶ 765] to provide reactive ADS extensions for accessing symbols that are loaded by the IAdsSymbolLoaderFactory

6.3.1 AdsClientExtensions Class

Extension class for AdsClient respective IAdsConnection [▶ 765] to provide reactive ADS extensions.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Reactive.AdsClientExtensions
Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class AdsClientExtensions
```

The AdsClientExtensions type exposes the following members.

Methods

	Name	Description
	PollAdsState(IAdsConnection, IObservable.Unit.) [▶ 1061]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

	Name	Description
	PollAdsState(IAdsConnection, TimeSpan) [▶ 1062]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	PollAdsStateAsync(IAdsConnection, IObservable.Unit., CancellationToken) [▶ 1064]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	PollAdsStateAsync(IAdsConnection, TimeSpan, CancellationToken) [▶ 1065]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s.
	WhenNotification(IAdsConnection, ISymbol) [▶ 1068]	Gets an observable sequence of Notification [▶ 974]s.
	WhenNotification(IAdsConnection, ISymbolCollection) [▶ 1069]	Gets an observable sequence of Notification [▶ 974] objects.
	WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [▶ 1070]	Gets an observable sequence of SymbolValueNotification [▶ 1104]s.
	WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings) [▶ 1071]	Gets an observable sequence of Notification [▶ 974] objects.
	WhenSymbolVersionChanges(IAdsConnection) [▶ 1073]	Gets an observable sequence of SymbolVersion changed counts.
	WhenSymbolVersionChanges(IAdsConnection, IScheduler) [▶ 1074]	Gets an observable sequence of SymbolVersion changed counts.

Remarks

Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced. (Beckhoff.TwinCAT.Ads.Reactive package on Nuget).

Examples

The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions

Observe changing ADS Symbols with reactive extensions.

```
// To Test the Observer run a project on the local PLC System (Port 851)

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    int eventCount = 1;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<SymbolValueNotification>(not =>
    {
        Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
    }
    );

    // Collect the symbols that are registered as Notification sources for their changed values.

    SymbolCollection notificationSymbols = new SymbolCollection();
    IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];

    foreach(ISymbol element in taskInfo.Elements)
    {
    ISymbol cycleCount = element.SubSymbols["CycleCount"];
    ISymbol lastExecTime = element.SubSymbols["LastExecTime"];

    notificationSymbols.Add(cycleCount);
    notificationSymbols.Add(lastExecTime);
    }

    // Create a subscription for the first 200 Notifications on Symbol Value changes.
    IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Defa
ult).Take(200).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

The following sample shows how observe AdsState [▶ 626] changed Notifications with the reactive AdsClientExtensions

Observe changing ADS states with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));
```

```
    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' --
> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.WhenAdsStateChanges().Buffer(2,1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

TwinCAT.Ads.Reactive Namespace [▶ 1056]

TwinCAT.Ads.Reactive.AnyTypeExtensions [▶ 1075]

TwinCAT.Ads.Reactive.ValueSymbolExtensions [▶ 1106]

6.3.1.1 AdsClientExtensions Methods

The AdsClientExtensions [▶ 1056] type exposes the following members.

Methods

	Name	Description
	PollAdsState(IAdsConnection, IObservable.Unit.) [▶ 1061]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	PollAdsState(IAdsConnection, TimeSpan) [▶ 1062]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	PollAdsStateAsync(IAdsConnection, IObservable.Unit., CancellationToken) [▶ 1064]	Gets an observable sequence of AdsState [▶ 626]s via Polling.
	PollAdsStateAsync(IAdsConnection, TimeSpan, CancellationToken) [▶ 1065]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

	Name	Description
	WhenAdsStateChanges [▶ 1066]	Gets an observable sequence of AdsState [▶ 626]s.
	WhenNotification(IAdsConnection, ISymbol) [▶ 1068]	Gets an observable sequence of Notification [▶ 974]s.
	WhenNotification(IAdsConnection, ISymbolCollection) [▶ 1069]	Gets an observable sequence of Notification [▶ 974] objects.
	WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [▶ 1070]	Gets an observable sequence of SymbolValueNotification [▶ 1104]s.
	WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings) [▶ 1071]	Gets an observable sequence of Notification [▶ 974] objects.
	WhenSymbolVersionChanges(IAdsConnection) [▶ 1073]	Gets an observable sequence of SymbolVersion changed counts.
	WhenSymbolVersionChanges(IAdsConnection, IScheduler) [▶ 1074]	Gets an observable sequence of SymbolVersion changed counts.

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.1.1.1 AdsClientExtensions.PollAdsState Method

Overload List

	Name	Description
	PollAdsState(IAdsConnection, IObservable.Unit.) [▶ 1061]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

	Name	Description
	PollAdsState(IAdsConnection, TimeSpan) [▶ 1062]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.PollAdsState Method (IAdsConnection, IObservable.Unit.)

Gets an observable sequence of AdsState [▶ 626]s via Polling.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<AdsState> PollAdsState(
    this IAdsConnection client,
    IObservable<Unit> trigger
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
trigger	Type: System.IObservable.Unit. The polling trigger

Return Value

Type: IObservable.AdsState [▶ 626].
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [▶ 626] via polling with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS States with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
```

```
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subs
cribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

PollAdsState Overload [▶ 1060]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [▶ 1066]

AdsClientExtensions.PollAdsState Method (IAdsConnection, TimeSpan)

Gets an observable sequence of AdsState [▶ 626]s via Polling.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<AdsState> PollAdsState(
    this IAdsConnection client,
    TimeSpan period
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
period	Type: System.TimeSpan The period.

Return Value

Type: IObservable.AdsState [▶ 626].
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [▶ 626] via polling with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS States with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subs
cribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

PollAdsState Overload [▶ 1060]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [▶ 1066]

6.3.1.1.2 AdsClientExtensions.PollAdsStateAsync Method

Overload List

	Name	Description
	PollAdsStateAsync(IAdsConnection, IObservable.Unit., CancellationToken) [▶ 1064]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

	Name	Description
	PollAdsStateAsync(IAdsConnection, TimeSpan, CancellationToken) [▶ 1065]	Gets an observable sequence of AdsState [▶ 626]s via Polling.

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.PollAdsStateAsync Method (IAdsConnection, IObservable.Unit., CancellationToken)

Gets an observable sequence of AdsState [▶ 626]s via Polling.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<AdsState> PollAdsStateAsync(
    this IAdsConnection client,
    IObservable<Unit> trigger,
    CancellationToken cancel
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
trigger	Type: System.IObservable.Unit. The polling trigger
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: IObservable.AdsState [▶ 626].
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [▶ 626] via polling with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS States with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subs
cribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

PollAdsStateAsync Overload [▶ 1063]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [▶ 1066]

AdsClientExtensions.PollAdsStateAsync Method (IAdsConnection, TimeSpan, CancellationToken)

Gets an observable sequence of AdsState [▶ 626]s via Polling.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<AdsState> PollAdsStateAsync(
    this IAdsConnection client,
    TimeSpan interval,
    CancellationToken cancel
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
interval	Type: System.TimeSpan The interval.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: IObservable.AdsState [▶ 626].
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [▶ 626] via polling with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS States with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Change Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
    AdsState oldValue = not[0];
    AdsState newValue = not[1];

    Console.WriteLine(string.Format("Changed ADSState from '{0}' --> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.PollAdsState(TimeSpan.FromMilliseconds(200)).Buffer(2, 1).Subs
cribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

PollAdsStateAsync Overload [▶ 1063]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenAdsStateChanges(IAdsConnection) [▶ 1066]

6.3.1.1.3 AdsClientExtensions.WhenAdsStateChanges Method

Gets an observable sequence of AdsState [▶ 626]s.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<AdsState> WhenAdsStateChanges(
    this IAdsConnection client
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.

Return Value

Type: IObservable.AdsState [▶ 626].
IObservable<AdsState>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe AdsState [▶ 626] changed Notifications with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS States with reactive extensions.

```
// To Test the observer, Start/Stop the local PLC

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.DefaultDynamic);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<IList<AdsState>>(not =>
    {
        AdsState oldValue = not[0];
        AdsState newValue = not[1];

        Console.WriteLine(string.Format("Changed ADSState from '{0}' --
> '{1}!", oldValue, newValue));
    }
    );

    // Create a subscription for the AdsState change and buffering 2 Values (for oldValue --
> newValue output).
    IDisposable subscription = client.WhenAdsStateChanges().Buffer(2,1).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.PollAdsState(IAdsConnection, IObservable.Unit.) [▶ 1061]

6.3.1.1.4 AdsClientExtensions.WhenNotification Method

Overload List

	Name	Description
	WhenNotification(IAdsConnection, ISymbol) [▶ 1068]	Gets an observable sequence of Notification [▶ 974]s.
	WhenNotification(IAdsConnection, ISymbolCollection) [▶ 1069]	Gets an observable sequence of Notification [▶ 974] objects.
	WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [▶ 1070]	Gets an observable sequence of SymbolValueNotification [▶ 1104]s.
	WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings) [▶ 1071]	Gets an observable sequence of Notification [▶ 974] objects.

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbol)

Gets an observable sequence of Notification [▶ 974]s.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Notification> WhenNotification(
    this IAdsConnection client,
    ISymbol symbol
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Return Value

Type: IObservable.Notification [▶ 974].
IObservable<NotificationValue>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsClientExtensions Class [▶ 1056]

WhenNotification Overload [▶ 1068]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings) [▶ 1071]

AdsClientExtensions.WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [▶ 1070]

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbolCollection)

Gets an observable sequence of Notification [▶ 974] objects.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<SymbolValueNotification> WhenNotification(
    this IAdsConnection connection,
    ISymbolCollection symbols
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
symbols	Type: TwinCAT.TypeSystem.ISymbolCollection [▶ 2182] The symbols.

Return Value

Type: IObservable.SymbolValueNotification [▶ 1104].
IObservable<NotificationValue>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS Symbols with reactive extensions.

```
// To Test the Observer run a project on the local PLC System (Port 851)

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    int eventCount = 1;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<SymbolValueNotification>(not =>
    {
        Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
    }
    );

    // Collect the symbols that are registered as Notification sources for their changed values.

    SymbolCollection notificationSymbols = new SymbolCollection();
    IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];

    foreach(ISymbol element in taskInfo.Elements)
    {
    ISymbol cycleCount = element.SubSymbols["CycleCount"];
    ISymbol lastExecTime = element.SubSymbols["LastExecTime"];

    notificationSymbols.Add(cycleCount);
    notificationSymbols.Add(lastExecTime);
    }

    // Create a subscription for the first 200 Notifications on Symbol Value changes.
    IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Defa
ult).Take(200).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

WhenNotification Overload [▶ 1068]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenNotification(IAdsConnection, ISymbol, NotificationSettings) [▶ 1070]

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbol, NotificationSettings)

Gets an observable sequence of SymbolValueNotification [▶ 1104]s.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<SymbolValueNotification> WhenNotification(
    this IAdsConnection client,
    ISymbol symbol,
    NotificationSettings settings
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol to observe.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] Notification settings.

Return Value

Type: IObservable.SymbolValueNotification [▶ 1104].
IObservable<NotificationValue>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Exceptions

Exception	Condition
ArgumentNullException	symbol
ArgumentOutOfRangeException	Symbol is not an IValueSymbol - symbol

Reference

AdsClientExtensions Class [▶ 1056]

WhenNotification Overload [▶ 1068]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenNotification(IAdsConnection, ISymbolCollection, NotificationSettings) [▶ 1071]

AdsClientExtensions.WhenNotification Method (IAdsConnection, ISymbolCollection, NotificationSettings)

Gets an observable sequence of Notification [▶ 974] objects.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<SymbolValueNotification> WhenNotification(
    this IAdsConnection client,
    ISymbolCollection symbols,
    NotificationSettings settings
)
```

Parameters

client	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
symbols	Type: TwinCAT.TypeSystem.ISymbolCollection [▶ 2182] The symbols to observe.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.

Return Value

Type: IObservable.SymbolValueNotification [▶ 1104].
IObservable<NotificationValue>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how observe Value changed Notifications with the reactive AdsClientExtensions [▶ 1056]

Observe changing ADS Symbols with reactive extensions.

```
// To Test the Observer run a project on the local PLC System (Port 851)

using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    int eventCount = 1;

    // Reactive Notification Handler
    var valueObserver = Observer.Create<SymbolValueNotification>(not =>
    {
        Console.WriteLine(string.Format("{0} {1:u} {2} = '{3}' ({4})", eventCount+
+, not.TimeStamp, not.Symbol.InstancePath, not.Value, not.Symbol.DataType));
    }
    );

    // Collect the symbols that are registered as Notification sources for their changed values.

    SymbolCollection notificationSymbols = new SymbolCollection();
    IArrayInstance taskInfo = (IArrayInstance)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo"];
```

```
    foreach(ISymbol element in taskInfo.Elements)
    {
    ISymbol cycleCount = element.SubSymbols["CycleCount"];
    ISymbol lastExecTime = element.SubSymbols["LastExecTime"];

    notificationSymbols.Add(cycleCount);
    notificationSymbols.Add(lastExecTime);
    }

    // Create a subscription for the first 200 Notifications on Symbol Value changes.
    IDisposable subscription = client.WhenNotification(notificationSymbols,NotificationSettings.Defa
ult).Take(200).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AdsClientExtensions Class [▶ 1056]

WhenNotification Overload [▶ 1068]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.1.1.5 AdsClientExtensions.WhenSymbolVersionChanges Method

Overload List

	Name	Description
	WhenSymbolVersionChanges(IAdsConnection) [▶ 1073]	Gets an observable sequence of SymbolVersion changed counts.
	WhenSymbolVersionChanges(IAdsConnection, IScheduler) [▶ 1074]	Gets an observable sequence of SymbolVersion changed counts.

Reference

AdsClientExtensions Class [▶ 1056]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenSymbolVersionChanges Method (IAdsConnection)

Gets an observable sequence of SymbolVersion changed counts.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<int> WhenSymbolVersionChanges(
    this IAdsConnection connection
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.

Return Value

Type: IObservable.Int32.
Counter, unique only within the WhenSymbolVersionChanges(IAdsConnection) observable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsClientExtensions Class [▶ 1056]

WhenSymbolVersionChanges Overload [▶ 1073]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AdsClientExtensions.WhenSymbolVersionChanges Method (IAdsConnection, IScheduler)

Gets an observable sequence of SymbolVersion changed counts.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<int> WhenSymbolVersionChanges(
    this IAdsConnection connection,
    IScheduler scheduler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The client.
scheduler	Type: IScheduler The scheduler.

Return Value

Type: IObservable.Int32.
Counter, unique only within the WhenSymbolVersionChanges(IAdsConnection, IScheduler) observable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AdsClientExtensions Class [▶ 1056]

WhenSymbolVersionChanges Overload [▶ 1073]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.2 AnyTypeExtensions Class

Extension class for IAdsConnection [▶ 765] to provide reactive ADS extensions (accessing symbol value sequences with the ANY_TYPE concept)

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Reactive.AnyTypeExtensions
Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class AnyTypeExtensions
```

Methods

	Name	Description
	PollValues(IAdsConnection, String, Type, IObservable.Unit.) [▶ 1089]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, TimeSpan) [▶ 1090]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan) [▶ 1093]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Polls the symbol values on timepoints where the polling observable streams data / triggers

	Name	Description
	PollValues(IAdsConnection, String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Polls the symbol values on time points where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, IObservable.Unit.) [▶ 1083]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan) [▶ 1084]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, .Int32., IObservable.Unit.) [▶ 1085]	Polls the symbol values on time points where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, .Int32., TimeSpan) [▶ 1086]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan, Func.Exception, T.) [▶ 1088]	Polls the symbol as value sequence of object values with a specified period time.

	Name	Description
	PollValues.T. (IAdsConnection, String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T. (IAdsConnection, String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Polls the symbol as value sequence of object values with a specified period time.
	WhenNotification(IAdsConnection, String, Type, NotificationSettings) [▶ 1099]	Creates an observable sequence of values that are created by ADS Notifications.
	WhenNotification.T. (IAdsConnection, String, NotificationSettings) [▶ 1098]	Creates an observable sequence of values that are created by ADS Notifications.
	WriteValues.T. (IAdsConnection, String, IObservable.T.) [▶ 1101]	Writes the sequence of values to the symbol specified by the instance path.
	WriteValues.T. (IAdsConnection, String, IObservable.T., Action.Exception.) [▶ 1102]	Writes the sequence of values to the symbol specified by the instance path.

Remarks

Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced. (Beckhoff.TwinCAT.Ads.Reactive package on Nuget).

Examples

Example1: Observe Value changed Notifications with the reactive AnyTypeExtensions

Observe a single changing ADS Symbols (Extended AdsNotifications, ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
```

```
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    }
    );

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

Example2: Polling ANY_TYPE values.

Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

Write values sequentially.

Write sequences of values to the target (ANY_TYPE)

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));
```

```
    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

TwinCAT.Ads.Reactive Namespace [▶ 1056]

TwinCAT.Ads.Reactive.AdsClientExtensions [▶ 1056]

TwinCAT.Ads.Reactive.ValueSymbolExtensions [▶ 1106]

6.3.2.1 AnyTypeExtensions Methods

Methods

	Name	Description
	PollValues(IAdsConnection, String, Type, IObservable.Unit.) [▶ 1089]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, TimeSpan) [▶ 1090]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan) [▶ 1093]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Polls the symbol values on time points where the polling observable streams data / triggers

	Name	Description
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, IObservable.Unit.) [▶ 1083]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan) [▶ 1084]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, .Int32., IObservable.Unit.) [▶ 1085]	Polls the symbol values on time points where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, .Int32., TimeSpan) [▶ 1086]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan, Func.Exception, T.) [▶ 1088]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Polls the symbol as value sequence of object values with a specified period time.

	Name	Description
	WhenNotification(IAdsConnection, String, Type, NotificationSettings) [▶ 1099]	Creates an observable sequence of values that are created by ADS Notifications.
	WhenNotification.T.(IAdsConnection, String, NotificationSettings) [▶ 1098]	Creates an observable sequence of values that are created by ADS Notifications.
	WriteValues.T.(IAdsConnection, String, IObservable.T.) [▶ 1101]	Writes the sequence of values to the symbol specified by the instance path.
	WriteValues.T.(IAdsConnection, String, IObservable.T., Action.Exception.) [▶ 1102]	Writes the sequence of values to the symbol specified by the instance path.

Reference

AnyTypeExtensions Class [▶ 1075]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.2.1.1 AnyTypeExtensions.PollValues Method

Overload List

	Name	Description
	PollValues.T.(IAdsConnection, String, IObservable.Unit.) [▶ 1083]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan) [▶ 1084]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, .Int32., IObservable.Unit.) [▶ 1085]	Polls the symbol values on time points where the polling observable streams data / triggers

	Name	Description
	PollValues.T.(IAdsConnection, String, .Int32., TimeSpan) [▶ 1086]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, IObservable.Unit., Func.Exception, T.) [▶ 1087]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, TimeSpan, Func.Exception, T.) [▶ 1088]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, IObservable.Unit.) [▶ 1089]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, TimeSpan) [▶ 1090]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues.T.(IAdsConnection, String, .Int32., IObservable.Unit., Func.Exception, T.) [▶ 1091]	Polls the symbol values on timepoints where the polling observable streams data / triggers
	PollValues.T.(IAdsConnection, String, .Int32., TimeSpan, Func.Exception, T.) [▶ 1092]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan) [▶ 1093]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, IObservable.Unit., Func.Exception, Object.) [▶ 1094]	Polls the symbol values on timepoints where the polling observable streams data / triggers

	Name	Description
	PollValues(IAdsConnection, String, Type, TimeSpan, Func.Exception, Object.) [▶ 1095]	Polls the symbol as value sequence of object values with a specified period time.
	PollValues(IAdsConnection, String, Type, .Int32., IObservable.Unit., Func.Exception, Object.) [▶ 1096]	Polls the symbol values on time points where the polling observable streams data / triggers
	PollValues(IAdsConnection, String, Type, .Int32., TimeSpan, Func.Exception, Object.) [▶ 1097]	Polls the symbol as value sequence of object values with a specified period time.

Reference

AnyTypeExtensions Class [▶ 1075]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, IObservable.Unit.)

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<Unit> trigger
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
trigger	Type: System.IObservable.Unit. The Polling trigger

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    TimeSpan period
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
period	Type: System.TimeSpan The period.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, .Int32., IObservable.Unit.)

Polls the symbol values on time points where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    IObservable<Unit> trigger
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
args	Type: .System.Int32. ANY_TYPE arguments
trigger	Type: System.IObservable.Unit. The Polling trigger

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, .Int32., TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    TimeSpan period
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
args	Type: .System.Int32. ANY_TYPE arguments.
period	Type: System.TimeSpan The period.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Polling ANY_TYPE values.

Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, IObservable.Unit., Func.Exception, T.)

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<Unit> trigger,
    Func<Exception, T> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
trigger	Type: System.IObservable.Unit. The Polling trigger
errorHandler	Type: System.Func.Exception, T. The error handler.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, TimeSpan, Func.Exception, T.)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    TimeSpan period,
    Func<Exception, T> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
period	Type: System.TimeSpan The period.
errorHandler	Type: System.Func.Exception, T. The error handler.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, IObservable.Unit.)

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    IObservable<Unit> trigger
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
trigger	Type: System.IObservable.Unit. The Polling trigger

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    TimeSpan period
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
period	Type: System.TimeSpan The period.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, .Int32., IObservable.Unit., Func.Exception, T.)

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    IObservable<Unit> trigger,
    Func<Exception, T> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
args	Type: .System.Int32. ANY_TYPE arguments
trigger	Type: System.IObservable.Unit. The Polling trigger
errorHandler	Type: System.Func.Exception, T. The error handler.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues.T. Method (IAdsConnection, String, .Int32., TimeSpan, Func.Exception, T.)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> PollValues<T>(
    this IAdsConnection connection,
    string instancePath,
    int[] args,
    TimeSpan period,
    Func<Exception, T> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
args	Type: .System.Int32. ANY_TYPE arguments.
period	Type: System.TimeSpan The period.
errorHandler	Type: System.Func.Exception, T. The error handler.

Type Parameters

T	The ANY_TYPE compatible .NET Type.

Return Value

Type: IObservable.T.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Polling ANY_TYPE values.

Observe changing ADS Symbols by polling (Read Polling) (ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
```

```
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., TimeSpan)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    TimeSpan period
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
args	Type: .System.Int32. The ANY_TYPE arguments.
period	Type: System.TimeSpan The period.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, IObservable.Unit., Func.Exception, Object.)

Polls the symbol values on timepoints where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    IObservable<Unit> trigger,
    Func<Exception, Object> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
trigger	Type: System.IObservable.Unit. The Polling trigger
errorHandler	Type: System.Func.Exception, Object. The error handler.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, TimeSpan, Func.Exception, Object.)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    TimeSpan period,
    Func<Exception, Object> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
period	Type: System.TimeSpan The period.
errorHandler	Type: System.Func.Exception, Object. The error handler.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., IObservable.Unit., Func.Exception, Object.)

Polls the symbol values on time points where the polling observable streams data / triggers

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    IObservable<Unit> trigger,
    Func<Exception, Object> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
args	Type: .System.Int32. The ANY_TYPE arguments.
trigger	Type: System.IObservable.Unit. The Polling trigger
errorHandler	Type: System.Func.Exception, Object. The error handler.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.PollValues Method (IAdsConnection, String, Type, .Int32., TimeSpan, Func.Exception, Object.)

Polls the symbol as value sequence of object values with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> PollValues(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    int[] args,
    TimeSpan period,
    Func<Exception, Object> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The data type of the symbol (ANY_TYPE)
args	Type: .System.Int32. The ANY_TYPE arguments.
period	Type: System.TimeSpan The period.
errorHandler	Type: System.Func.Exception, Object. The error handler.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

AnyTypeExtensions Class [▶ 1075]

PollValues Overload [▶ 1081]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.2.1.2 AnyTypeExtensions.WhenNotification Method

Overload List

	Name	Description
	WhenNotification.T.(IAdsConnection, String, NotificationSettings) [▶ 1098]	Creates an observable sequence of values that are created by ADS Notifications.
	WhenNotification(IAdsConnection, String, Type, NotificationSettings) [▶ 1099]	Creates an observable sequence of values that are created by ADS Notifications.

Reference

AnyTypeExtensions Class [▶ 1075]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.WhenNotification.T. Method (IAdsConnection, String, NotificationSettings)

Creates an observable sequence of values that are created by ADS Notifications.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<T> WhenNotification<T>(
    this IAdsConnection connection,
    string instancePath,
    NotificationSettings settings
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.

Type Parameters

T	The .NET Type representation of the specified symbols type.

Return Value

Type: IObservable.T.
IObservable<T>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Remarks

The values will be cast to the specified type. The .NET type must fit the Symbol type like all ANYTYPES.

Examples

The following sample shows how to observe Value changed Notifications with the reactive AnyTypeExtensions [▶ 1075]

Observe changing ADS Symbols with reactive extensions (Extended AdsNotification, ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    }
    );

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

WhenNotification Overload [▶ 1098]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.WhenNotification Method (IAdsConnection, String, Type, NotificationSettings)

Creates an observable sequence of values that are created by ADS Notifications.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> WhenNotification(
    this IAdsConnection connection,
    string instancePath,
    Type type,
    NotificationSettings settings
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
type	Type: System.Type The type.
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The settings.

Return Value

Type: IObservable.Object.
IObservable<T>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Remarks

The values will be cast to the specified type. The .NET type must fit be one of the compatible 'ANYTYPES'.

Examples

The following sample shows how to observe Value changed Notifications with the reactive AnyTypeExtensions [▶ 1075]

Observe changing ADS Symbols with reactive extensions (Extended AdsNotifications, ANY_TYPE)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Reactive Notification Handler
    var valueObserver = Observer.Create<ushort>(val =>
    {
    Console.WriteLine(string.Format("Value: {0}", val.ToString()));
    }
    );

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenNotification<ushort>("TwinCAT_SystemInfoVarList._TaskInfo.
CycleCount", NotificationSettings.Default).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

WhenNotification Overload [▶ 1098]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.2.1.3 AnyTypeExtensions.WriteValues Method

Overload List

	Name	Description
	WriteValues.T. (IAdsConnection, String, IObservable.T.) [▶ 1101]	Writes the sequence of values to the symbol specified by the instance path.
	WriteValues.T. (IAdsConnection, String, IObservable.T., Action.Exception.) [▶ 1102]	Writes the sequence of values to the symbol specified by the instance path.

Reference

AnyTypeExtensions Class [▶ 1075]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.WriteValues.T. Method (IAdsConnection, String, IObservable.T.)

Writes the sequence of values to the symbol specified by the instance path.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IDisposable WriteValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<T> valueSequence
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.

valueSequence	Type: System.IObservable.T. Value sequence (Any type).

Type Parameters

T

Return Value

Type: IDisposable
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Write values sequentially.

Write sequences of values to the target (ANY_TYPE)

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

WriteValues Overload [▶ 1101]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

AnyTypeExtensions.WriteValues.T. Method (IAdsConnection, String, IObservable.T., Action.Exception.)

Writes the sequence of values to the symbol specified by the instance path.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IDisposable WriteValues<T>(
    this IAdsConnection connection,
    string instancePath,
    IObservable<T> valueSequence,
    Action<Exception> errorHandler
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePath	Type: System.String The instance path.
valueSequence	Type: System.IObservable.T. Value sequence (Any type).
errorHandler	Type: System.Action.Exception. The error handler.

Type Parameters

T

Return Value

Type: IDisposable
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

Write values sequentially.

Write sequences of values to the target (ANY_TYPE)

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol gvlIntSymbol = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object (short) Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = gvlIntSymbol.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

AnyTypeExtensions Class [▶ 1075]

WriteValues Overload [▶ 1101]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.3 SymbolValueNotification Class

Symbol Notification class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Notification [▶ 974]
    ValueNotification
      TwinCAT.Ads.Reactive.SymbolValueNotification

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class SymbolValueNotification : ValueNotification
```

The SymbolValueNotification type exposes the following members.

Properties

	Name	Description
	Data [▶ 977]	The notification Data. (Inherited from Notification [▶ 974].)
	Handle [▶ 977]	The notification handle (Inherited from Notification [▶ 974].)
	Symbol [▶ 1105]	Gets the symbol of the SymbolValueNotification.
	TimeStamp [▶ 978]	Gets the time stamp of the INotification [▶ 969] (Inherited from Notification [▶ 974].)
	UserData [▶ 978]	Attached UserData/Tag at the INotification [▶ 969] (Inherited from Notification [▶ 974].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

Extends the ValueNotification class by symbol specific information.

Reference

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.3.1 SymbolValueNotification Properties

The SymbolValueNotification [▶ 1104] type exposes the following members.

Properties

	Name	Description
	Data [▶ 977]	The notification Data. (Inherited from Notification [▶ 974].)
	Handle [▶ 977]	The notification handle (Inherited from Notification [▶ 974].)
	Symbol [▶ 1105]	Gets the symbol of the SymbolValueNotification [▶ 1104].
	TimeStamp [▶ 978]	Gets the time stamp of the INotification [▶ 969] (Inherited from Notification [▶ 974].)
	UserData [▶ 978]	Attached UserData/Tag at the INotification [▶ 969] (Inherited from Notification [▶ 974].)

Reference

SymbolValueNotification Class [▶ 1104]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.3.1.1 SymbolValueNotification.Symbol Property

Gets the symbol of the SymbolValueNotification [▶ 1104].

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Symbol { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The value symbol.

Reference

SymbolValueNotification Class [▶ 1104]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.3.2 SymbolValueNotification Methods

The SymbolValueNotification [▶ 1104] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SymbolValueNotification Class [▶ 1104]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.4 ValueSymbolExtensions Class

Extension class for IAdsConnection [▶ 765] to provide reactive ADS extensions for accessing symbols that are loaded by the IAdsSymbolLoaderFactory

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Reactive.ValueSymbolExtensions
Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class ValueSymbolExtensions
```

Methods

	Name	Description
	PollValuesAnnotated(IValueSymbol, IObservable.Unit.) [▶ 1111]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence
	PollValuesAnnotated(IValueSymbol, TimeSpan) [▶ 1112]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time.
	WhenValueChanged(IValueSymbol) [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed.

	Name	Description
	WhenValueChanged (IAdsConnection, IEnumerable.ISymbol.) [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol.
	WriteValues(IValueSymbol, IObservable.Object.) [▶ 1117]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception.) [▶ 1118]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., CancellationToken) [▶ 1119]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Remarks

Reactive Extensions (Rx) are a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators. Using Rx, developers represent asynchronous data streams with Observables, query asynchronous data streams using LINQ operators, and parameterize the concurrency in the asynchronous data streams using Schedulers. Simply put, Rx = Observables + LINQ + Schedulers. The ADS reactive extensions are build on top of this library to enable ADS Symbol and State Observables, seamlessly bound to the reactive extensions. To use the ADS reactive extensions the TwinCAT.Ads.Reactive Nuget package (or the included TwinCAT.Ads.Reactive.dll) must be referenced from All types within are contained in the ADS companion package "Beckhoff.TwinCAT.Ads.Reactive" which must be referenced separately. (Beckhoff.TwinCAT.Ads.Reactive package on Nuget).

Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions from an IValueSymbol [▶ 2254].

Observe a single changing ADS Symbol (ADS Notifications)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
```

```
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions from an **DynamicSymbol**.

Observe a single changing ADS Symbol (ADS Notifications) with the dynamic language runtime (.NET DLR)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

The same for more than one IValueSymbol [▶ 2254].

Observe changing ADS Symbols (ADS Notifications)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"]; // UShort Type
```

```
    IValueSymbol lastExecTime = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo.LastExecTime"]; // UInt Type

    SymbolCollection symbols = new SymbolCollection();
    symbols.Add(cycleCount);
    symbols.Add(lastExecTime);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange,500,5000); // o
ptional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenValueChanged(symbols).Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

Here, the values are polled in a specific time period and sequential Reads are triggered (in opposite to ADS Notification in the latter example)

Observe changing ADS Symbols by polling (Read Polling)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    // Take 20 Values in an Interval of 500ms
    IDisposable subscription = cycleCount.PollValues(TimeSpan.FromMilliseconds(500)).Take(20).Subscr
ibe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

In the following example it is demonstrated how to write Values sequentially to a IValueSymbol [▶ 2254] with the help of the reactive extensions.

Write sequences of values to the target

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["GVL.i"];
```

```
    // Produces object Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = cycleCount.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

TwinCAT.Ads.Reactive Namespace [▶ 1056]

TwinCAT.Ads.Reactive.AdsClientExtensions [▶ 1056]

TwinCAT.Ads.Reactive.AnyTypeExtensions [▶ 1075]

6.3.4.1 ValueSymbolExtensions Methods

Methods

	Name	Description
	PollValuesAnnotated(IValueSymbol, IObservable.Unit.) [▶ 1111]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence
	PollValuesAnnotated(IValueSymbol, TimeSpan) [▶ 1112]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time.
	WhenValueChanged(IValueSymbol) [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed.
	WhenValueChanged(IAdsConnection, IEnumerable.ISymbol.) [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol.
	WriteValues(IValueSymbol, IObservable.Object.) [▶ 1117]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception.) [▶ 1118]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

	Name	Description
	WriteValues(IValueSymbol, IObservable.Object., CancellationToken) [▶ 1119]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Reference

ValueSymbolExtensions Class [▶ 1106]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.4.1.1 ValueSymbolExtensions.PollValuesAnnotated Method

Overload List

	Name	Description
	PollValuesAnnotated(IValueSymbol, IObservable.Unit.) [▶ 1111]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence
	PollValuesAnnotated(IValueSymbol, TimeSpan) [▶ 1112]	Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time.

Reference

ValueSymbolExtensions Class [▶ 1106]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.PollValuesAnnotated Method (IValueSymbol, IObservable.Unit.)

Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<ValueChangedEventArgs> PollValuesAnnotated(
    this IValueSymbol symbol,
    IObservable<Unit> trigger
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
trigger	Type: System.IObservable.Unit. The polling Trigger.

Return Value

Type: IObservable.ValueChangedEventArgs [▶ 2439].
IObservable<ValueChangedArgs>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

ValueSymbolExtensions Class [▶ 1106]

PollValuesAnnotated Overload [▶ 1111]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.PollValuesAnnotated Method (IValueSymbol, TimeSpan)

Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<ValueChangedEventArgs> PollValuesAnnotated(
    this IValueSymbol symbol,
    TimeSpan period
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
period	Type: System.TimeSpan The polling period/interval.

Return Value

Type: IObservable.ValueChangedEventArgs [▶ 2439].
IObservable<ValueChangedArgs>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

ValueSymbolExtensions Class [▶ 1106]

PollValuesAnnotated Overload [▶ 1111]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.3.4.1.2 ValueSymbolExtensions.WhenValueChanged Method

Overload List

	Name	Description
	WhenValueChanged (IValueSymbol) [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed.
	WhenValueChanged (IAdsConnection, IEnumerable.ISymbol.) [▶ 1114]	Observable sequence of Values driven by ADS Notifications on the specified symbol.

Reference

ValueSymbolExtensions Class [▶ 1106]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WhenValueChanged Method (IValueSymbol)

Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> WhenValueChanged(
    this IValueSymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.

Return Value

Type: IObservable.Object.
IObservable<System.Object>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions [▶ 1106] from an IValueSymbol [▶ 2254].

Observe a single changing ADS Symbols (ADS Notifications)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class [▶ 1106]

WhenValueChanged Overload [▶ 1113]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WhenValueChanged Method (IAdsConnection, IEnumerable.ISymbol.)

Observable sequence of Values driven by ADS Notifications on the specified symbol.

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IObservable<Object> WhenValueChanged(
    this IAdsConnection connection,
    IEnumerable<ISymbol> symbols
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The ADS connection / ADS Client
symbols	Type: System.Collections.Generic.IEnumerable.ISymbol [▶ 2176]. The symbols to observe.

Return Value

Type: IObservable.Object.
IObservable<ValueChangedArgs>.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IAdsConnection [▶ 765]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

The same for more than one IValueSymbol [▶ 2254].

Observe changing ADS Symbols (ADS Notifications)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o.CycleCount"]; // UShort Type
    IValueSymbol lastExecTime = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskI
nfo.LastExecTime"]; // UInt Type

    SymbolCollection symbols = new SymbolCollection();
    symbols.Add(cycleCount);
    symbols.Add(lastExecTime);

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange,500,5000); // o
ptional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = client.WhenValueChanged(symbols).Take(20).Subscribe(valueObserver);
```

```
    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Examples

The following sample shows how to observe Value changed Notifications with the reactive ValueSymbolExtensions [▶ 1106] from an **DynamicSymbol**.

Observe a single changing ADS Symbol (ADS Notifications) with the dynamic language runtime (.NET DLR)

```
// To Test the Observer run a project on the local PLC System (Port 851)
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["TwinCAT_SystemInfoVarList._TaskInf
o[1].CycleCount"];

    // Reactive Notification Handler
    var valueObserver = Observer.Create<object>(val =>
    {
    Console.WriteLine(string.Format("Instance: {0}, Value: {1}", cycleCount.InstancePath, val.ToStri
ng()));
    }
    );

    cycleCount.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 500, 5000); //
 optional: Change NotificationSettings on Symbol

    // Turning ADS Notifications into sequences of Value Objects (Taking 20 Values)
    // and subscribe to them.
    IDisposable subscription = cycleCount.WhenValueChanged().Take(20).Subscribe(valueObserver);

    Console.ReadKey(); // Wait for Key press
    subscription.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class [▶ 1106]

WhenValueChanged Overload [▶ 1113]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

Observable

6.3.4.1.3 ValueSymbolExtensions.WriteValues Method

Overload List

	Name	Description
	WriteValues(IValueSymbol, IObservable.Object.) [▶ 1117]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

	Name	Description
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception.) [▶ 1118]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., CancellationToken) [▶ 1119]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].
	WriteValues(IValueSymbol, IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Reference

ValueSymbolExtensions Class [▶ 1106]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable.Object.)

Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IDisposable WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
valueObservable	Type: System.IObservable.Object. Observable of Values.

Return Value

Type: IDisposable
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Examples

In the following example it is demonstrated how to write Values sequentially to a IValueSymbol [▶ 2254] with the help of the reactive extensions.

Write sequences of values to the target

```
using (AdsClient client = new AdsClient())
{
    // Connect to target
    client.Connect(new AmsAddress(AmsNetId.Local, 851));

    // Create Symbol information (Symbol 'i : INT' in PLC Global Variables list.
    var symbolLoader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    IValueSymbol cycleCount = (IValueSymbol)symbolLoader.Symbols["GVL.i"];

    // Produces object Values 0,1,2,3 ... in seconds period
    IObservable<object> timerObservable = Observable.Interval(TimeSpan.FromSeconds(1.0)).Select(i =>
 (object)(short)i);

    // Take 10 Values (0..9) and write them to GVL.i
    IDisposable dispose = cycleCount.WriteValues(timerObservable.Take(10));

    Console.ReadKey(); // Wait for Key press
    dispose.Dispose(); // Dispose the Subscription
}
```

Reference

ValueSymbolExtensions Class [▶ 1106]

WriteValues Overload [▶ 1116]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable.Object., Action.Exception.)

Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IDisposable WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable,
    Action<Exception> errorHandler
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.

valueObservable	Type: System.IObservable.Object. Observable of Values.
errorHandler	Type: System.Action.Exception. The error handler or NULL.

Return Value

Type: IDisposable
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

ValueSymbolExtensions Class [▶ 1106]

WriteValues Overload [▶ 1116]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable.Object., CancellationToken)

Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static void WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
valueObservable	Type: System.IObservable.Object. Observable of Values.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type:
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

ValueSymbolExtensions Class [▶ 1106]

WriteValues Overload [▶ 1116]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

ValueSymbolExtensions.WriteValues Method (IValueSymbol, IObservable.Object., Action.Exception., CancellationToken)

Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254].

Namespace: TwinCAT.Ads.Reactive [▶ 1056]
Assembly: TwinCAT.Ads.Reactive (in TwinCAT.Ads.Reactive.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static void WriteValues(
    this IValueSymbol symbol,
    IObservable<Object> valueObservable,
    Action<Exception> errorHandler,
    CancellationToken cancel
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
valueObservable	Type: System.IObservable.Object. Observable of Values.
errorHandler	Type: System.Action.Exception. The error handler.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type:
IDisposable.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IValueSymbol [▶ 2254]. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

ValueSymbolExtensions Class [▶ 1106]

WriteValues Overload [▶ 1116]

TwinCAT.Ads.Reactive Namespace [▶ 1056]

6.4 TwinCAT.Ads.Server Namespace

Namespace for the TwinCAT ADS Server Component.

Classes

	Class	Description
	AdsServer [▶ 1121]	Base implementation for an ADS Server.
	AdsServerException [▶ 1181]	An AdsServerException [▶ 1181] is thrown on communication errors in the AdsServer [▶ 1121] class.
	ErrorEventArgs [▶ 1186]	This class implements the event arguments passed by the TcAdsServerExEvent.
	LoopbackNotRegisteredException [▶ 1188]	The Tcp Loopback client is not registered. Implements the AdsServerException [▶ 1181]
	NotificationDataSample [▶ 1191]	This class implements an ADS Notification Sample. It contains the notification handle and the variable data.
	NotificationSamplesStamp [▶ 1195]	This class implements an ADS Stamp Header containing multiple ADS Notification Samples (TcAdsStampHeader)
	ServerConnectionStateChangedEventArgs [▶ 1199]	Class ConnectionStateChangedEventArgs (Server Connections)
	ServerNotConnectedException [▶ 1202]	The AdsServer is not connected. Implements the AdsServerException [▶ 1181]

Enumerations

	Enumeration	Description
	ServerConnectionState [▶ 1198]	The Server Connection State

6.4.1 AdsServer Class

Base implementation for an ADS Server.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Server.AdsServer
Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class AdsServer : IDisposable
```

The AdsServer type exposes the following members.

Properties

	Name	Description
	AmsServer [▶ 1135]	Gets the the internal AmsServer [▶ 1135] object.
	IsConnected [▶ 1136]	Gets a value indicating whether AdsServer is connected.
	IsDisconnecting [▶ 1136]	Indicates, that the AdsServer is actually disconnecting.
	IsDisposed [▶ 1137]	Gets a value indicating whether this instance is disposed.
	Logger [▶ 1137]	Gets the logger object.
	ServerAddress [▶ 1138]	The AMS address of this server.

Methods

	Name	Description
	AddDeviceNotificationConfirmationAsync [▶ 1142]	Called when an ADS Add Device Notification confirmation is received.
	AddDeviceNotificationIndicationAsync [▶ 1143]	Called when an ADS Add Device Notification indication is received.
	AddDeviceNotificationRequest [▶ 1144]	Sends an ADS Add Device Notification request (synchronous).
	AddDeviceNotificationRequestAsync [▶ 1144]	Sends an ADS Add Device Notification request (async)
	AddDeviceNotificationResponseAsync [▶ 1145]	Sends an ADS Add Device Notification response.
	ConnectServer [▶ 1146]	Connect this ADS server to the local ADS router.

	Name	Description
	ConnectServerAndWaitAsync [▶ 1147]	Registers the AdsServer at the router asynchronously.
	DeleteDeviceNotificationConfirmationAsync [▶ 1147]	Called when an ADS Delete Device Notification confirmation is received.
	DeleteDeviceNotificationIndicationAsync [▶ 1148]	Called when an ADS Delete Device Notification indication is received.
	DeleteDeviceNotificationRequest [▶ 1149]	Sends an ADS Delete Device Notification request (synchronous).
	DeleteDeviceNotificationRequestAsync [▶ 1149]	Sends an ADS Delete Device Notification request (async).
	DeleteDeviceNotificationResponseAsync [▶ 1150]	Sends an ADS Delete Device Notification response.
	DeviceNotificationRequestAsync [▶ 1151]	Sends an ADS Device Notification request asynchronously
	DeviceNotificationRequestSync [▶ 1152]	Sends an ADS Device Notification request (sync)
	Disconnect [▶ 1152]	Disconnects this ADS server from the local ADS router.
	Dispose. [▶ 1153]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 1154]	Releases unmanaged and - optionally - managed resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 1154]	Finalizes an instance of the AdsServer class. (Overrides Object.Finalize.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnAddDeviceNotificationConfirmationAsync [▶ 1155]	Called when Add device Notification Confirmation is send.
	OnConnected [▶ 1155]	Handler function that is called, when the AdsServer is connected.
	OnDisconnect [▶ 1156]	Called when the AdsServer is about to be disconnected.

	Name	Description
	OnRouterNotification [▶ 1156]	Handler Function for a Router Notification.
	OnServerConnectionStateChanged [▶ 1157]	Handles the ServerConnectionStateChanged [▶ 1181] event.
	ReadConfirmationAsync [▶ 1157]	
	ReadDeviceInfoConfirmationAsync [▶ 1158]	Called when an ADS Read Device Info confirmation is received.
	ReadDeviceInfoIndicationAsync [▶ 1159]	Called when an ADS Read Device Info indication is received.
	ReadDeviceInfoRequestAsync [▶ 1160]	Sends an ADS Read Device Info request asynchronously
	ReadDeviceInfoRequestSync [▶ 1160]	Sends an ADS Read Device Info request synchronously.
	ReadDeviceInfoResponseAsync [▶ 1161]	Sends an ADS Read Device Info response.
	ReadDeviceStateConfirmationAsync [▶ 1162]	Called when an ADS Read State confirmation is received.
	ReadDeviceStateIndicationAsync [▶ 1163]	Called when an ADS Read State indication is received.
	ReadDeviceStateRequestAsync [▶ 1163]	Sends an ADS Read State request (asynchronous)
	ReadDeviceStateRequestSync [▶ 1164]	Sends an ADS Read State request (synchronous)
	ReadDeviceStateResponseAsync [▶ 1165]	Sends an ADS Read State response.
	ReadIndicationAsync [▶ 1166]	Called when an ADS Read indication is received.
	ReadRequest [▶ 1166]	Sends an ADS Read Request.
	ReadRequestAsync [▶ 1167]	Sends an ADS Read Request asynchronously.

	Name	Description
	ReadResponseAsync [▶ 1168]	
	ReadWriteConfirmationAsync [▶ 1169]	
	ReadWriteIndicationAsync [▶ 1169]	
	ReadWriteRequestAsync [▶ 1170]	
	ReadWriteRequestSync [▶ 1171]	
	ReadWriteResponseAsync [▶ 1172]	
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	WriteConfirmationAsync [▶ 1172]	Called when an ADS Write confirmation is received. Overwrite this method in derived classes to react on ADS Write confirmations.
	WriteControlConfirmationAsync [▶ 1173]	Called when an ADS Write Control confirmation is received.
	WriteControlIndicationAsync [▶ 1174]	
	WriteControlRequest [▶ 1175]	
	WriteControlRequestAsync [▶ 1175]	
	WriteControlRequestSync [▶ 1176]	
	WriteControlResponseAsync [▶ 1177]	Sends an ADS Write Control response.
	WriteIndicationAsync [▶ 1178]	
	WriteRequest [▶ 1178]	
	WriteRequestAsync [▶ 1179]	
	WriteResponseAsync [▶ 1180]	Sends an ADS Write response.

Events

	Name	Description
⚡	ServerConnectionStateChanged [▶ 1181]	The connection status has changed

Remarks

Derived classes should overwrite the indication methods to react on incoming requests. The confirmation methods should be overwritten to receive replies on asynchronous requests sent by this ADS server.

Examples

The following sample shows how to derive from the AdsServer class and create your own Customized ADS Server.

AdsSampleServer

C#

```
class Program
{
    public static void Main(string[] args)
    {
    CreateHostBuilder(args).Build().Run();
    }

    public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args)
        .ConfigureServices((hostContext, services) =>
        {
        services.AddHostedService<ServerWorker>();
        });
}
```

C#

```
public class ServerWorker : BackgroundService
{
    private readonly ILogger<ServerWorker> _logger;

    public ServerWorker(ILogger<ServerWorker> logger)
    {
    _logger = logger;
    }

    protected override async Task ExecuteAsync(CancellationToken cancel)
    {
    // Instantiate the server
    AdsSampleServer server = new AdsSampleServer(_logger);
    // Connect the server and wait for cancel
    await server.ConnectServerAndWaitAsync(cancel);
    }
}
```

C#

```
/*
 * Extend the AdsServer class to implement your own ADS server.
 */
public class AdsSampleServer : AdsServer
{
    /// <summary>
    /// Fixed ADS Port (to be changed ...)
    /// </summary>
    const ushort ADS_PORT = 42;

    /// <summary>
    /// Fixed Name for the ADS Port (change this ...)
```

```
    /// </summary>
    const string ADS_PORT_NAME = "AdsSampleServer_Port42";

    /// <summary>
    /// Some simple data / ProcessImage
    /// </summary>
    private byte[] _dataBuffer = {1, 2, 3, 4};

    /// <summary>
    /// Ads State
    /// </summary>
    private AdsState _adsState = AdsState.Config;
    /// <summary>
    /// Device State
    /// </summary>
    private ushort _deviceState = 0;

    /// <summary>
    /// Notification dictionary, thread safe
    /// </summary>
    private ConcurrentDictionary<uint, NotificationRequestEntry> _notificationTable = new Concurrent
Dictionary<uint, NotificationRequestEntry>();

    /// <summary>
    /// Simple counter for different Notification handles here.
    /// </summary>
    private uint _currentNotificationHandle = 0;

    /// <summary>
    /// Logger
    /// </summary>
    private ILogger _logger;

    /* Instanstiate an ADS server with a fix ADS port assigned by the ADS router.
    */

    public AdsSampleServer()
    : this(null)
    {
    }

    public AdsSampleServer(ILogger logger) : base(ADS_PORT, ADS_PORT_NAME)
    {
    _logger = logger;
    }

    /// <summary>
    /// Trace log message
    /// </summary>
    /// <param name="message">The message.</param>
    /// <param name="args">The arguments.</param>
    private void LogTrace(string message, params object[] args)
    {
    if (_logger != null)
        _logger.LogTrace(message, args);
    }

    /// <summary>
    /// Information Log
    /// </summary>
    /// <param name="message">The message.</param>
    /// <param name="args">The arguments.</param>
    private void LogInformation(string message, params object[] args)
    {
    if (_logger != null)
        _logger.LogInformation(message, args);
    }

    /// <summary>
    /// Error log
    /// </summary>
    /// <param name="message">The message.</param>
    /// <param name="args">The arguments.</param>
    private void LogError(string message, params object[] args)
    {
    if (_logger != null)
        _logger.LogError(message, args);
    }

    /// <summary>
```

```
    /// AdsServer Version
    /// </summary>
    static AdsVersion s_version =  new AdsVersion(0,0,1);

    /* Overwrite the indication methods of the TcAdsServer class for the services your ADS server
     * provides. They are called upon incoming requests. All indications that are not overwritten in
     * this class return the ADS DeviceServiceNotSupported error code to the requester.
     */

    /// <summary>
    /// Called when an ADS Read Device Info indication is received by your <see cref="AdsSampleServe
r"/>.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read Device Info indications.
    /// The default implementation replies with an <see AdsErrorCode.DeviceServiceNotSupported></
see> error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's / requester's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="OnReadDeviceInfoIndicationAsync"
/> operation. The <see cref="Task{T}"/> parameter contains the <see cref="AdsErrorCode"/> as
    /// <see cref="Task{Task}.Result"/>.
    /// </returns>
    protected override Task<AdsErrorCode> ReadDeviceInfoIndicationAsync(AmsAddress sender, uint invo
keId, CancellationToken cancel)
    {
    LogTrace("ReadDeviceINfoIndication(Address:{0}, ID: {1})", sender, invokeId);

    // Send a response to the requester
    return ReadDeviceInfoResponseAsync(sender,   // requester's AMS address
        invokeId,                   // invoke id provided by requester
        AdsErrorCode.NoError,           // ADS error code
        "C#_TestServer",            // name of this server
        s_version,                  // version of this server
        cancel);                // Cancellation Token
    }

    /// <summary>
    /// Called when an ADS Write indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Write indications.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="indexGroup">The index group of the requested ADS service</param>
    /// <param name="indexOffset">The index offset of the requested ADS service</param>
    /// <param name="writeData">The data to be written</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="WriteIndicationAsync" /
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> WriteIndicationAsync(AmsAddress sender, uint invokeId, uin
t indexGroup, uint indexOffset, ReadOnlyMemory<byte> writeData, CancellationToken cancel)
    {
    AdsErrorCode adsError = AdsErrorCode.NoError;

    LogTrace("WriteIndicationAsync(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:
{4})", sender, invokeId, indexGroup, indexOffset, writeData.Length);

    switch (indexGroup) /* use index group (and offset) to distinguish between the services
                of this server */
    {
        case 0x10000:
        if (writeData.Length == 4)
        {
            writeData.CopyTo(_dataBuffer);
        }
        else
        {
            adsError = AdsErrorCode.DeviceInvalidParam;
        }

        break;
        case 0x20000: /* used for the PLC Sample */
        if (writeData.Length == 4)
        {
```

```
            uint value = BinaryPrimitives.ReadUInt32LittleEndian(writeData.Span);
            LogInformation(String.Format("PLC Counter: {0}", value));
        }

        break;

        default: /* other services are not supported */
        adsError = AdsErrorCode.DeviceServiceNotSupported;
        break;
    }

    // Send a response to the requester

    return WriteResponseAsync(sender, // requester's AMS address
        invokeId, // invoke id provided by requester
        adsError, // ADS error code
        cancel);
    }

    /// <summary>
    /// Called when an ADS Read indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read indications.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="indexGroup">The index group of the requested ADS service</param>
    /// <param name="indexOffset">The index offset of the requested ADS service</param>
    /// <param name="readLength">The number of bytes to be read</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadIndicationAsync"/
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> ReadIndicationAsync(AmsAddress sender, uint invokeId, uint
 indexGroup, uint indexOffset, int readLength, CancellationToken cancel)
    {
    LogTrace("ReadIndicationAsync(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:
{4})", sender, invokeId, indexGroup, indexOffset, readLength);

    /* Distinguish between services like in AdsWriteInd */

    // Send a response to the requester
    return ReadResponseAsync(sender, // requester's AMS address
        invokeId,           // invoke id provided by requester
        AdsErrorCode.NoError,       // ADS error code
        _dataBuffer.AsMemory(),     // data buffer
        cancel);
    }

    /// <summary>
    /// Called when an ADS Read State indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read State indications.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadDeviceStateIndicationAsync"
/> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> ReadDeviceStateIndicationAsync(AmsAddress sender, uint inv
okeId, CancellationToken cancel)
    {
    LogTrace("ReadIndicationAsync(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:
{4})", sender, invokeId);

    return ReadDeviceStateResponseAsync(sender, // requester's AMS address
        invokeId,                   // invoke id provided by requester
        AdsErrorCode.NoError,           // ADS error code
        _adsState,                  // ADS state
        _deviceState,               // device state
        cancel);
    }

    /// <summary>
    /// Called when an ADS Write Control indication is received.
```

```
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Write Control indications.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="adsState">The requested new ADS state of this ADS device</param>
    /// <param name="deviceState">The requested new device state of this ADS device</param>
    /// <param name="cbLength">The length in bytes of the additional data buffer</param>
    /// <param name="data">An additional data buffer of cbLength bytes</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="WriteControlIndicationAsync"/
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> WriteControlIndicationAsync(AmsAddress sender, uint invoke
Id, AdsState adsState, ushort deviceState, ReadOnlyMemory<byte> data, CancellationToken cancel)
    {
    LogTrace("WriteControlIndication(Address:{0}, ID:{1}, AdsState:{2}, DeviceState:{3}, Length:
{4})", sender, invokeId, adsState, deviceState, data.Length);

    // Set requested ADS and device status

    _adsState = adsState;
    _deviceState = deviceState;

    // Send a response to the requester

    return WriteControlResponseAsync(sender,    // requester's AMS address
        invokeId,                   // invoke id provided by requester
        AdsErrorCode.NoError,           // ADS error code
        cancel);
    }

    /// <summary>
    /// Called when an ADS Add Device Notification indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Add Device Notification indications
.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="indexGroup">The index group of the requested ADS service</param>
    /// <param name="indexOffset">The index offset of the requested ADS service</param>
    /// <param name="dataLength">Number of bytes to be transmitted</param>
    /// <param name="settings">The Notification settings.</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous 'AddDeviceNotificationIndication' operation
. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> AddDeviceNotificationIndicationAsync(AmsAddress sender, ui
nt invokeId, uint indexGroup, uint indexOffset, int dataLength, NotificationSettings settings, Cance
llationToken cancel)
    {
    LogTrace("AddDeviceNotificationIndication(Address:{0}, ID:{1}, IG:{2}, IO:{3}, Length:
{4})", sender, invokeId, indexGroup, indexOffset, dataLength);

    /* Create a new notification entry an store it in the notification table */
    NotificationRequestEntry notEntry = new NotificationRequestEntry(sender, indexGroup, indexOffset
, dataLength, settings);

    _notificationTable.AddOrUpdate(_currentNotificationHandle, notEntry,(key,value) => notEntry);
    _currentNotificationHandle++;

    // Send a response to the requester
    return AddDeviceNotificationResponseAsync(sender, invokeId, AdsErrorCode.NoError, _currentNotifi
cationHandle++, cancel);
    }

    /// <summary>
    /// Called when an ADS Delete Device Notification indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Delete Device Notification indicati
ons.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
```

```
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="hNotification">The notification handle to be deleted</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="DeleteDeviceNotificationIndicati
onAsync"/> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /
> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> DeleteDeviceNotificationIndicationAsync(AmsAddress sender,
 uint invokeId, uint hNotification, CancellationToken cancel)
    {
    LogTrace("DeleteDeviceNotification(Address:{0}, ID:{1}, Handle:
{2})", sender, invokeId, hNotification);

    AdsErrorCode errorCode = AdsErrorCode.NoError;

    /* check if the requested notification handle is still in the notification table */
    if (_notificationTable.ContainsKey(hNotification))
    {
        NotificationRequestEntry entry = null;
        _notificationTable.TryRemove(hNotification, out entry);
    }
    else // notification handle is not in the notification table -> return an error code
        // to the requester
    {
        errorCode = AdsErrorCode.DeviceNotifyHandleInvalid;
    }

    // Send a response to the requester
    return DeleteDeviceNotificationResponseAsync(sender, invokeId, errorCode, cancel);
    }

    /// <summary>
    /// Called when an ADS Device Notification indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Device Notification indications.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="numStampHeaders">The number of ADS Stamp Headers contained in stampHeaders</
param>
    /// <param name="stampHeaders">The array of received ADS Stamp Headers.</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="DeviceNotificationIndicationAsyn
c(AmsAddress, uint, uint, NotificationSamplesStamp[], CancellationToken)"/
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    /// <exclude/>
    protected override Task<AdsErrorCode> DeviceNotificationIndicationAsync(AmsAddress sender, uint
invokeId, uint numStampHeaders, NotificationSamplesStamp[] stampHeaders, CancellationToken cancel)
    {
    LogTrace("DeviceNotificationIndication(Address:{0}, ID:{1}, NumStampHeaders:
{2})", sender, invokeId, numStampHeaders);
    LogInformation("Received Device Notification Request");

    /*
     * Call notification handlers.
     */
    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Read Write indication is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read Write indications.
    /// The default implementation replies with an ADS ServiceNotSupported error code (0x701).
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invokeId provided by the sender</param>
    /// <param name="indexGroup">The index group of the requested ADS service</param>
    /// <param name="indexOffset">The index offset of the requested ADS service</param>
    /// <param name="readLength">Number of bytes to be read</param>
    /// <param name="writeData">The data to be written</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous 'ReadWriteIndication' operation. The <see c
ref="Task{T}"/> parameter contains the <see cref="AdsErrorCode"/> as
    /// <see cref="Task{Task}.Result"/>.
    /// </returns>
    protected async override Task<AdsErrorCode> ReadWriteIndicationAsync(AmsAddress sender, uint inv
```

```
okeId, uint indexGroup, uint indexOffset, int readLength, ReadOnlyMemory<byte> writeData, Cancellati
onToken cancel)
    {
    LogTrace("ReadWriteIndication(Address:{0}, ID:{1}, IG:{2}, IO:{3}, ReadLen:{4}, WriteLen:
{5})", sender, invokeId, indexGroup, indexOffset, readLength,writeData.Length);

    /* Distinguish between services like in AdsWriteInd */
    // Send a response to the requester

    AdsErrorCode errorCode = AdsErrorCode.None;

    if (readLength == 4 && writeData.Length == 4)
    {
        errorCode = await ReadWriteResponseAsync(sender, // requester's AMS address
        invokeId, // invoke id provided by requester
        AdsErrorCode.NoError, // ADS error code
        _dataBuffer.AsMemory(), cancel).ConfigureAwait(false);
        writeData.CopyTo(_dataBuffer.AsMemory(0, 4));
    }
    else
    {
        errorCode = await ReadWriteResponseAsync(sender, // requester's AMS address
        invokeId, // invoke id provided by requester
        AdsErrorCode.DeviceInvalidSize, // ADS error code
        Memory<byte>.Empty, cancel).ConfigureAwait(false);
    }

    return errorCode;
    }

    /* Overwrite the  confirmation methods of the TcAdsServer class for the requests your ADS server
     * sends. They are called upon incoming responses. These sample implementations only add a log m
essage
     * to the sample form.
    */

    /// <summary>
    /// Called when an ADS Read State confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read State confirmations.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="adsState">The ADS state of the sender</param>
    /// <param name="deviceState">The device state of the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadDeviceStateConfirmationAsync
" /> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> ReadDeviceStateConfirmationAsync(AmsAddress sender, uint i
nvokeId, AdsErrorCode result, AdsState adsState, ushort deviceState, CancellationToken cancel)
    {
    LogTrace("ReadDeviceStateConfirmation(Address:{0}, ID:{1}, Result:{2}, AdsState:
{3}, DeviceSTate:{4})", sender, invokeId, result, adsState, deviceState);
    LogInformation("Received Read State Confirmation");
    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Read confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read confirmations.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="cbLength">The number of read bytes</param>
    /// <param name="data">The read data buffer</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadConfirmationAsync"/
> operation. The <see cref="Task{T}"/> parameter contains the <see cref="AdsErrorCode"/> as
    /// <see cref="Task{Task}.Result"/>.
    /// </returns>
    protected override Task<AdsErrorCode> ReadConfirmationAsync(AmsAddress sender, uint invokeId, Ad
sErrorCode result, ReadOnlyMemory<byte> readData, CancellationToken cancel)
```

```
    {
    LogTrace("ReadConfirmation(Address:{0}, ID:{1}, IG:{2}, Result:{3}, Length:
{4})", sender, invokeId, result, readData.Length);
    LogInformation("Received Read Confirmation");

    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Write confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Write confirmations.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="WriteConfirmationAsync" /
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> WriteConfirmationAsync(AmsAddress sender, uint invokeId, A
dsErrorCode result, CancellationToken cancel)
    {
    LogTrace("WriteConfirmation(Address:{0}, ID:{1}, Result:{2})", sender, invokeId, result);
    LogInformation("Received Write Confirmation");

    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Read Device Info confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read Device Info confirmations.
    /// </remarks>
    /// <param name="target">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="name">The sender's name</param>
    /// <param name="version">The sender's version</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadDeviceInfoConfirmationAsync(
AmsAddress, uint, AdsErrorCode, string, AdsVersion, CancellationToken)" /
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> ReadDeviceInfoConfirmationAsync(AmsAddress sender, uint in
vokeId, AdsErrorCode result, string name, AdsVersion version, CancellationToken cancel)
    {
    LogTrace("ReadDeviceInfoConfirmation(Address:{0}, ID:{1}, Result:{2}, Name:{3}, Version:
{4})", sender, invokeId, result, name, version);
    LogInformation("Received Read Device Info Confirmation");

    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Write Control confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Write Control confirmations.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="WriteControlConfirmationAsync" /
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> WriteControlConfirmationAsync(AmsAddress sender, uint invo
keId, AdsErrorCode result, CancellationToken cancel)
    {
    LogTrace("WriteControlConfirmation(Address:{0}, ID:{1}, Result:{2})", sender, invokeId, result);
    LogInformation("Received Write Control Confirmation");

    return Task.FromResult(AdsErrorCode.Succeeded);
    }
```

```
    /// <summary>
    /// Called when an ADS Add Device Notification confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Add Device Notification confirmatio
ns.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="notificationHandle">The notification handle provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="AddDeviceNotificationConfirmatio
nAsync" /> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /
> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> AddDeviceNotificationConfirmationAsync(AmsAddress sender,
uint invokeId, AdsErrorCode result, uint notificationHandle, CancellationToken cancel)
    {
    //_serverLogger.ServerNotificationHandle = notificationHandle;

    LogTrace("AddDeviceNotificationConfirmation(Address:{0}, ID:{1}, Result:{2}, Handle:
{3})", sender, invokeId, result,notificationHandle);
    LogInformation("Received Add Device Notification Confirmation. Notification handle: " + notifica
tionHandle);

    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Delete Device Notification confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Delete Device Notification confirma
tions.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="DeleteDeviceNotificationConfirma
tionAsync" /> operation. The <see cref="Task{T}" /
> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> DeleteDeviceNotificationConfirmationAsync(AmsAddress sende
r, uint invokeId, AdsErrorCode result, CancellationToken cancel)
    {
    LogTrace("DeleteDeviceNotificationConfirmation(Address:{0}, ID:{1}, Result:
{2})", sender, invokeId, result);
    LogInformation("Received Delete Device Notification Confirmation");

    return Task.FromResult(AdsErrorCode.Succeeded);
    }

    /// <summary>
    /// Called when an ADS Read Write confirmation is received.
    /// </summary>
    /// <remarks>
    /// Overwrite this method in derived classes to react on ADS Read Write confirmations.
    /// </remarks>
    /// <param name="sender">The sender's AMS address</param>
    /// <param name="invokeId">The invoke id provided by this server during the corresponding reques
t</param>
    /// <param name="result">The ADS error code provided by the sender</param>
    /// <param name="cbLength">The  number of read bytes</param>
    /// <param name="data">The read data buffer</param>
    /// <param name="cancel">The cancellation token.</param>
    /// <returns>A task that represents the asynchronous <see cref="ReadWriteConfirmationAsync" /
> operation. The <see cref="Task{T}" /> parameter contains the <see cref="AdsErrorCode" /> as
    /// <see cref="Task{Task}.Result" />.</returns>
    protected override Task<AdsErrorCode> ReadWriteConfirmationAsync(AmsAddress sender, uint invokeI
d, AdsErrorCode result, ReadOnlyMemory<byte> readData, CancellationToken cancel)
    {
    LogTrace("AddDeviceNotificationIndication(Address:{0}, ID:{1}, Result:{2}, Length:
{3})", sender, invokeId, result, readData.Length);
    LogInformation("Received Read Write Confirmation");
    return Task.FromResult(AdsErrorCode.Succeeded);
```

```
    }
}

/// <summary>
/// AdsSampleServer Notification request entry
/// </summary>
internal class NotificationRequestEntry
{
    private AmsAddress _rAddr;      // the AmsNetId of the requester
    private uint _indexGroup;       // the requested index group
    private uint _indexOffset;      // the requested index offset
    private int _cbLength;     // the number of bytes to send
    NotificationSettings _settings; // the notification settings

    internal NotificationRequestEntry(AmsAddress rAddr,
                      uint indexGroup,
                      uint indexOffset,
                      int cbLength,
                      NotificationSettings settings)
    {
    _rAddr = rAddr;
    _indexGroup = indexGroup;
    _indexOffset = indexOffset;
    _cbLength = cbLength;
    _settings = settings;
    }
}
```

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1 AdsServer Properties

The AdsServer [▶ 1121] type exposes the following members.

Properties

	Name	Description
	AmsServer [▶ 1135]	Gets the the internal AmsServer [▶ 1135] object.
	IsConnected [▶ 1136]	Gets a value indicating whether AdsServer [▶ 1121] is connected.
	IsDisconnecting [▶ 1136]	Indicates, that the AdsServer [▶ 1121] is actually disconnecting.
	IsDisposed [▶ 1137]	Gets a value indicating whether this instance is disposed.
	Logger [▶ 1137]	Gets the logger object.
	ServerAddress [▶ 1138]	The AMS address of this server.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.1 AdsServer.AmsServer Property

Gets the the internal AmsServer object.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AmsServer AmsServer { get; }
```

Property Value

Type: AmsServer
The ams server.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.2 AdsServer.IsConnected Property

Gets a value indicating whether AdsServer [▶ 1121] is connected.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsConnected { get; }
```

Property Value

Type: Boolean
true if this instance is connected; otherwise, false.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.3 AdsServer.IsDisconnecting Property

Indicates, that the AdsServer [▶ 1121] is actually disconnecting.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected bool IsDisconnecting { get; }
```

Property Value

Type: Boolean

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.4 AdsServer.IsDisposed Property

Gets a value indicating whether this instance is disposed.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsDisposed { get; }
```

Property Value

Type: Boolean
true if this instance is disposed; otherwise, false.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.5 AdsServer.Logger Property

Gets the logger object.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected ILogger Logger { get; }
```

Property Value

Type: ILogger
The logger.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.1.6 AdsServer.ServerAddress Property

The AMS address of this server.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual AmsAddress ServerAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]

Remarks

The Address consists of AmsNetId [▶ 665] and AmsPort [▶ 693]. While the AmsNetId [▶ 665] is defined by the System, where the **AmsTcpIpRouter** is running, the AmsPort [▶ 693] is specified by the AdsServer [▶ 1121] constructor. By default, the router is running on the same system, but can be configured by RouterEndPoint [▶ 1546].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

TwinCAT.Ads.AmsNetId [▶ 665]

TwinCAT.Ads.AmsPort [▶ 693]

AmsConfiguration.RouterEndPoint [▶ 1546]

6.4.1.2 AdsServer Methods

The AdsServer [▶ 1121] type exposes the following members.

Methods

	Name	Description
	AddDeviceNotificationConfirmationAsync [▶ 1142]	Called when an ADS Add Device Notification confirmation is received.
	AddDeviceNotificationIndicationAsync [▶ 1143]	Called when an ADS Add Device Notification indication is received.
	AddDeviceNotificationRequest [▶ 1144]	Sends an ADS Add Device Notification request (synchronous).

	Name	Description
	AddDeviceNotificationRequestAsync [▶ 1144]	Sends an ADS Add Device Notification request (async)
	AddDeviceNotificationResponseAsync [▶ 1145]	Sends an ADS Add Device Notification response.
	ConnectServer [▶ 1146]	Connect this ADS server to the local ADS router.
	ConnectServerAndWaitAsync [▶ 1147]	Registers the AdsServer [▶ 1121] at the router asynchronously.
	DeleteDeviceNotificationConfirmationAsync [▶ 1147]	Called when an ADS Delete Device Notification confirmation is received.
	DeleteDeviceNotificationIndicationAsync [▶ 1148]	Called when an ADS Delete Device Notification indication is received.
	DeleteDeviceNotificationRequest [▶ 1149]	Sends an ADS Delete Device Notification request (synchronous).
	DeleteDeviceNotificationRequestAsync [▶ 1149]	Sends an ADS Delete Device Notification request (async).
	DeleteDeviceNotificationResponseAsync [▶ 1150]	Sends an ADS Delete Device Notification response.
	DeviceNotificationRequestAsync [▶ 1151]	Sends an ADS Device Notification request asynchronously
	DeviceNotificationRequestSync [▶ 1152]	Sends an ADS Device Notification request (sync)
	Disconnect [▶ 1152]	Disconnects this ADS server from the local ADS router.
	Dispose. [▶ 1153]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 1154]	Releases unmanaged and - optionally - managed resources.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize [▶ 1154]	Finalizes an instance of the AdsServer [▶ 1121] class. (Overrides Object.Finalize..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnAddDeviceNotificationConfirmationAsync [▶ 1155]	Called when Add device Notification Confirmation is send.
	OnConnected [▶ 1155]	Handler function that is called, when the AdsServer [▶ 1121] is connected.
	OnDisconnect [▶ 1156]	Called when the AdsServer [▶ 1121] is about to be disconnected.
	OnRouterNotification [▶ 1156]	Handler Function for a Router Notification.
	OnServerConnectionStateChanged [▶ 1157]	Handles the ServerConnectionStateChanged [▶ 1181] event.
	ReadConfirmationAsync [▶ 1157]	
	ReadDeviceInfoConfirmationAsync [▶ 1158]	Called when an ADS Read Device Info confirmation is received.
	ReadDeviceInfoIndicationAsync [▶ 1159]	Called when an ADS Read Device Info indication is received.
	ReadDeviceInfoRequestAsync [▶ 1160]	Sends an ADS Read Device Info request asynchronously
	ReadDeviceInfoRequestSync [▶ 1160]	Sends an ADS Read Device Info request synchronously.
	ReadDeviceInfoResponseAsync [▶ 1161]	Sends an ADS Read Device Info response.
	ReadDeviceStateConfirmationAsync [▶ 1162]	Called when an ADS Read State confirmation is received.
	ReadDeviceStateIndicationAsync [▶ 1163]	Called when an ADS Read State indication is received.
	ReadDeviceStateRequestAsync [▶ 1163]	Sends an ADS Read State request (asynchronous)
	ReadDeviceStateRequestSync [▶ 1164]	Sends an ADS Read State request (synchronous)
	ReadDeviceStateResponseAsync [▶ 1165]	Sends an ADS Read State response.

	Name	Description
	ReadIndicationAsync [▶ 1166]	Called when an ADS Read indication is received.
	ReadRequest [▶ 1166]	Sends an ADS Read Request.
	ReadRequestAsync [▶ 1167]	Sends an ADS Read Request asynchronously.
	ReadResponseAsync [▶ 1168]	
	ReadWriteConfirmationAsync [▶ 1169]	
	ReadWriteIndicationAsync [▶ 1169]	
	ReadWriteRequestAsync [▶ 1170]	
	ReadWriteRequestSync [▶ 1171]	
	ReadWriteResponseAsync [▶ 1172]	
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	WriteConfirmationAsync [▶ 1172]	Called when an ADS Write confirmation is received. Overwrite this method in derived classes to react on ADS Write confirmations.
	WriteControlConfirmationAsync [▶ 1173]	Called when an ADS Write Control confirmation is received.
	WriteControlIndicationAsync [▶ 1174]	
	WriteControlRequest [▶ 1175]	
	WriteControlRequestAsync [▶ 1175]	
	WriteControlRequestSync [▶ 1176]	
	WriteControlResponseAsync [▶ 1177]	Sends an ADS Write Control response.
	WriteIndicationAsync [▶ 1178]	

	Name	Description
	WriteRequest [▶ 1178]	
	WriteRequestAsync [▶ 1179]	
	WriteResponseAsync [▶ 1180]	Sends an ADS Write response.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.1 AdsServer.AddDeviceNotificationConfirmationAsync Method

Called when an ADS Add Device Notification confirmation is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> AddDeviceNotificationConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
notificationHandle	Type: System.UInt32 The notification handle provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous AddDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Add Device Notification confirmations.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.2 AdsServer.AddDeviceNotificationIndicationAsync Method

Called when an ADS Add Device Notification indication is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> AddDeviceNotificationIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invokeId provided by the sender
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index offset of the requested ADS service
dataLength	Type: System.Int32 Number of bytes to be transmitted
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The Notification settings.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous 'AddDeviceNotificationIndication' operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Add Device Notification indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.3 AdsServer.AddDeviceNotificationRequest Method

Sends an ADS Add Device Notification request (synchronous).

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode AddDeviceNotificationRequest(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index group of the requested ADS service
dataLength	Type: System.Int32 The number of bytes to be transmitted
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.4 AdsServer.AddDeviceNotificationRequestAsync Method

Sends an ADS Add Device Notification request (async)

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> AddDeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int dataLength,
    NotificationSettings settings,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index group of the requested ADS service
dataLength	Type: System.Int32 The number of bytes to be transmitted
settings	Type: TwinCAT.Ads.NotificationSettings [▶ 979] The notification settings.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous AddDeviceNotificationRequestAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, NotificationSettings, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.5 AdsServer.AddDeviceNotificationResponseAsync Method

Sends an ADS Add Device Notification response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> AddDeviceNotificationResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
```

```
    uint handle,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
handle	Type: System.UInt32 The notification handle for the added notification
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous AddDeviceNotificationResponseAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.6 AdsServer.ConnectServer Method

Connect this ADS server to the local ADS router.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual uint ConnectServer()
```

Return Value

Type: UInt32
The AmsServer Port.

Exceptions

Exception	Condition
AdsServerException [▶ 1181]	The connect call has failed!.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.7 AdsServer.ConnectServerAndWaitAsync Method

Registers the AdsServer [▶ 1121] at the router asynchronously.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<AdsErrorCode> ConnectServerAndWaitAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task.AdsErrorCode [▶ 575].
Returns a task object that represents the ConnectServerAndWaitAsync(CancellationToken) operation which returns an AdsErrorCode [▶ 575]" as result..

Remarks

The connection is hold until a cancel is requested, that means this method will wait until disconnect.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.8 AdsServer.DeleteDeviceNotificationConfirmationAsync Method

Called when an ADS Delete Device Notification confirmation is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> DeleteDeviceNotificationConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous DeleteDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Delete Device Notification confirmations.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.9 AdsServer.DeleteDeviceNotificationIndicationAsync Method

Called when an ADS Delete Device Notification indication is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> DeleteDeviceNotificationIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint hNotification,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invokeId provided by the sender
hNotification	Type: System.UInt32 The notification handle to be deleted
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous DeleteDeviceNotificationIndicationAsync(AmsAddress, UInt32, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Delete Device Notification indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.10 AdsServer.DeleteDeviceNotificationRequest Method

Sends an ADS Delete Device Notification request (synchronous).

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode DeleteDeviceNotificationRequest(
    AmsAddress target,
    uint invokeId,
    uint hNotification
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
hNotification	Type: System.UInt32 The notification ID to be deleted

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.11 AdsServer.DeleteDeviceNotificationRequestAsync Method

Sends an ADS Delete Device Notification request (async).

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> DeleteDeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint hNotification,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
hNotification	Type: System.UInt32 The notification ID to be deleted
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous DeleteDeviceNotificationRequestAsync(AmsAddress, UInt32, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.12 AdsServer.DeleteDeviceNotificationResponseAsync Method

Sends an ADS Delete Device Notification response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> DeleteDeviceNotificationResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous DeleteDeviceNotificationResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.13 AdsServer.DeviceNotificationRequestAsync Method

Sends an ADS Device Notification request asynchronously

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> DeviceNotificationRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint numStampHeaders,
    NotificationSamplesStamp[] notificationHeaders,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
numStampHeaders	Type: System.UInt32 The number of ADS Stamp Headers to be sent
notificationHeaders	Type: .TwinCAT.Ads.Server.NotificationSamplesStamp [▶ 1195]. The array of ADS Stamp Headers to be sent
cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task.AdsErrorCode [▶ 575].
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.14 AdsServer.DeviceNotificationRequestSync Method

Sends an ADS Device Notification request (sync)

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode DeviceNotificationRequestSync(
    AmsAddress target,
    uint invokeId,
    uint numStampHeaders,
    NotificationSamplesStamp[] notificationHeaders
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
numStampHeaders	Type: System.UInt32 The number of ADS Stamp Headers to be sent
notificationHeaders	Type: .TwinCAT.Ads.Server.NotificationSamplesStamp [▶ 1195]. The array of ADS Stamp Headers to be sent

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.15 AdsServer.Disconnect Method

Disconnects this ADS server from the local ADS router.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool Disconnect()
```

Return Value

Type: Boolean
true if disconnected, false if the AdsServer [▶ 1121] was diconnected before.

Exceptions

Exception	Condition
AdsServerException [▶ 1181]	Thrown if the disconnect call fails.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.16 AdsServer.Dispose Method

Overload List

	Name	Description
	Dispose. [▶ 1153]	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean) [▶ 1154]	Releases unmanaged and - optionally - managed resources.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

AdsServer.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Dispose()
```

Implements

IDisposable.Dispose.

Reference

AdsServer Class [▶ 1121]

Dispose Overload [▶ 1153]

TwinCAT.Ads.Server Namespace [▶ 1121]

AdsServer.Dispose Method (Boolean)

Releases unmanaged and - optionally - managed resources.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void Dispose(
    bool disposing
)
```

Parameters

disposing	Type: System.Boolean true to release both managed and unmanaged resources; false to release only unmanaged resources.

Reference

AdsServer Class [▶ 1121]

Dispose Overload [▶ 1153]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.17 AdsServer.Finalize Method

Finalizes an instance of the AdsServer [▶ 1121] class.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override void Finalize()
```

Implements

Object.Finalize.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.18 AdsServer.OnAddDeviceNotificationConfirmationAsync Method

Called when Add device Notification Confirmation is send.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> OnAddDeviceNotificationConfirmationAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    uint notificationHandle,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The r addr.
invokeId	Type: System.UInt32 The invoke identifier.
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The result.
notificationHandle	Type: System.UInt32 The notification handle.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous OnAddDeviceNotificationConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.19 AdsServer.OnConnected Method

Handler function that is called, when the AdsServer [▶ 1121] is connected.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnConnected()
```

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.20 AdsServer.OnDisconnect Method

Called when the AdsServer [▶ 1121] is about to be disconnected.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual bool OnDisconnect()
```

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.21 AdsServer.OnRouterNotification Method

Handler Function for a Router Notification.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual void OnRouterNotification(
    AmsRouterState state
)
```

Parameters

state	Type: TwinCAT.Ads.AmsRouterState [▶ 697] The route state.

Exceptions

Exception	Condition
NotImplementedException	

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.22 AdsServer.OnServerConnectionStateChanged Method

Handles the ServerConnectionStateChanged [▶ 1181] event.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnServerConnectionStateChanged(
    Object sender,
    ServerConnectionStateChangedEventArgs e
)
```

Parameters

sender	Type: System.Object The sender.
e	Type: TwinCAT.Ads.Server.ServerConnectionStateChangedEventArgs [▶ 1199] The ServerConnectionStateChangedEventArgs [▶ 1199] instance containing the event data.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.23 AdsServer.ReadConfirmationAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadConfirmationAsync(
    AmsAddress targetAddress,
    uint invokeId,
    AdsErrorCode result,
```

```
    ReadOnlyMemory readData,
    void cancel
)
```

Parameters

targetAddress	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575]
readData	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.24 AdsServer.ReadDeviceInfoConfirmationAsync Method

Called when an ADS Read Device Info confirmation is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadDeviceInfoConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    string name,
    AdsVersion version,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
name	Type: System.String The sender's name

version	Type: TwinCAT.Ads.AdsVersion [▶ 642] The sender's version
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceInfoConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, String, AdsVersion, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read Device Info confirmations.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.25 AdsServer.ReadDeviceInfoIndicationAsync Method

Called when an ADS Read Device Info indication is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadDeviceInfoIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's / requester's AMS address
invokeId	Type: System.UInt32 The invokeId provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous OnReadDeviceInfoIndicationAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read Device Info indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.26 AdsServer.ReadDeviceInfoRequestAsync Method

Sends an ADS Read Device Info request asynchronously

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadDeviceInfoRequestAsync(
    AmsAddress target,
    uint invokeId,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceInfoRequestAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.27 AdsServer.ReadDeviceInfoRequestSync Method

Sends an ADS Read Device Info request synchronously.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode ReadDeviceInfoRequestSync(
    AmsAddress target,
    uint invokeId
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.28 AdsServer.ReadDeviceInfoResponseAsync Method

Sends an ADS Read Device Info response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadDeviceInfoResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    string name,
    AdsVersion version,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
name	Type: System.String The name of this ADS server
version	Type: TwinCAT.Ads.AdsVersion [▶ 642] The version of this ADS server

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceInfoResponseAsync(AmsAddress, UInt32, AdsErrorCode, String, AdsVersion, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.29 AdsServer.ReadDeviceStateConfirmationAsync Method

Called when an ADS Read State confirmation is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadDeviceStateConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    AdsState adsState,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
adsState	Type: TwinCAT.Ads.AdsState [▶ 626] The ADS state of the sender
deviceState	Type: System.UInt16 The device state of the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceStateConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, AdsState, UInt16, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read State confirmations.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.30 AdsServer.ReadDeviceStateIndicationAsync Method

Called when an ADS Read State indication is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadDeviceStateIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invokeId provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceStateIndicationAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read State indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.31 AdsServer.ReadDeviceStateRequestAsync Method

Sends an ADS Read State request (asynchronous)

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadDeviceStateRequestAsync(
    AmsAddress target,
    uint invokeId,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceStateRequestAsync(AmsAddress, UInt32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.32 AdsServer.ReadDeviceStateRequestSync Method

Sends an ADS Read State request (synchronous)

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode ReadDeviceStateRequestSync(
    AmsAddress target,
    uint invokeId
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.33 AdsServer.ReadDeviceStateResponseAsync Method

Sends an ADS Read State response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadDeviceStateResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    AdsState adsState,
    ushort deviceState,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
adsState	Type: TwinCAT.Ads.AdsState [▶ 626] The current ADS state of this ADS server
deviceState	Type: System.UInt16 The device state of this ADS server
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadDeviceStateResponseAsync(AmsAddress, UInt32, AdsErrorCode, AdsState, UInt16, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.34 AdsServer.ReadIndicationAsync Method

Called when an ADS Read indication is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invokeId provided by the sender
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index offset of the requested ADS service
readLength	Type: System.Int32 The number of bytes to be read
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadIndicationAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Read indications. The default implementation replies with an ADS ServiceNotSupported error code (0x701).

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.35 AdsServer.ReadRequest Method

Sends an ADS Read Request.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode ReadRequest(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index group of the requested ADS service
readLength	Type: System.Int32 The number of bytes to be read

Return Value

Type: AdsErrorCode [▶ 575]
The ADS error code for this call.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.36 AdsServer.ReadRequestAsync Method

Sends an ADS Read Request asynchronously.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address

invokeId	Type: System.UInt32 The invokeId for this call. Used to find the matching ADS Confirmation
indexGroup	Type: System.UInt32 The index group of the requested ADS service
indexOffset	Type: System.UInt32 The index group of the requested ADS service
readLength	Type: System.Int32 The number of bytes to be read
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous ReadRequestAsync(AmsAddress, UInt32, UInt32, UInt32, Int32, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.37 AdsServer.ReadResponseAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    Memory data,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575]
data	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.38 AdsServer.ReadWriteConfirmationAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadWriteConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    ReadOnlyMemory readData,
    void cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575]
readData	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.39 AdsServer.ReadWriteIndicationAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> ReadWriteIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlyMemory writeData,
    void cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readLength	Type: System.Int32
writeData	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.40 AdsServer.ReadWriteRequestAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadWriteRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlySpan writeData,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readLength	Type: System.Int32
writeData	Type: ReadOnlySpan
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.41 AdsServer.ReadWriteRequestSync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode ReadWriteRequestSync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    int readLength,
    ReadOnlySpan data
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
readLength	Type: System.Int32

data	Type: ReadOnlySpan

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.42 AdsServer.ReadWriteResponseAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> ReadWriteResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    ReadOnlyMemory data,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575]
data	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.43 AdsServer.WriteConfirmationAsync Method

Called when an ADS Write confirmation is received. Overwrite this method in derived classes to react on ADS Write confirmations.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> WriteConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address
invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous WriteConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.44 AdsServer.WriteControlConfirmationAsync Method

Called when an ADS Write Control confirmation is received.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> WriteControlConfirmationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648] The sender's AMS address

invokeId	Type: System.UInt32 The invoke id provided by this server during the corresponding request
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code provided by the sender
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous WriteControlConfirmationAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Remarks

Overwrite this method in derived classes to react on ADS Write Control confirmations.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.45 AdsServer.WriteControlIndicationAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> WriteControlIndicationAsync(
    AmsAddress sender,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlyMemory data,
    void cancel
)
```

Parameters

sender	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
data	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.46 AdsServer.WriteControlRequest Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode WriteControlRequest(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
data	Type: ReadOnlySpan

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.47 AdsServer.WriteControlRequestAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> WriteControlRequestAsync(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
data	Type: ReadOnlySpan
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.48 AdsServer.WriteControlRequestSync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode WriteControlRequestSync(
    AmsAddress target,
    uint invokeId,
    AdsState adsState,
    ushort deviceState,
    ReadOnlySpan data
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]

invokeId	Type: System.UInt32
adsState	Type: TwinCAT.Ads.AdsState [▶ 626]
deviceState	Type: System.UInt16
data	Type: ReadOnlySpan

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.49 AdsServer.WriteControlResponseAsync Method

Sends an ADS Write Control response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> WriteControlResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous WriteControlResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.50 AdsServer.WriteIndicationAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<AdsErrorCode> WriteIndicationAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    ReadOnlyMemory writeData,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
writeData	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.51 AdsServer.WriteRequest Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsErrorCode WriteRequest(
    AmsAddress target,
    uint invokeId,
```

```
    uint indexGroup,
    uint indexOffset,
    ReadOnlySpan data
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
data	Type: ReadOnlySpan

Return Value

Type: AdsErrorCode [▶ 575]

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.52 AdsServer.WriteRequestAsync Method

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> WriteRequestAsync(
    AmsAddress target,
    uint invokeId,
    uint indexGroup,
    uint indexOffset,
    ReadOnlySpan data,
    void cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648]
invokeId	Type: System.UInt32
indexGroup	Type: System.UInt32
indexOffset	Type: System.UInt32
data	Type: ReadOnlySpan

cancel	Type: System.Void

Return Value

Type: Task.AdsErrorCode [▶ 575].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.2.53 AdsServer.WriteResponseAsync Method

Sends an ADS Write response.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Task<AdsErrorCode> WriteResponseAsync(
    AmsAddress target,
    uint invokeId,
    AdsErrorCode result,
    CancellationToken cancel
)
```

Parameters

target	Type: TwinCAT.Ads.AmsAddress [▶ 648] The receiver's AMS address
invokeId	Type: System.UInt32 The invoke ID provided by the receiver
result	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The ADS error code for the response
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.AdsErrorCode [▶ 575].
A task that represents the asynchronous WriteResponseAsync(AmsAddress, UInt32, AdsErrorCode, CancellationToken) operation. The Task.TResult. parameter contains the AdsErrorCode [▶ 575] as Result.

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.3 AdsServer Events

The AdsServer [▶ 1121] type exposes the following members.

Events

	Name	Description
	ServerConnectionStateChanged [▶ 1181]	The connection status has changed

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.1.3.1 AdsServer.ServerConnectionStateChanged Event

The connection status has changed

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ServerConnectionStateChangedEventArgs> ServerConnectionStateChanged
```

Value

Type: System.EventHandler.ServerConnectionStateChangedEventArgs [▶ 1199].

Reference

AdsServer Class [▶ 1121]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2 AdsServerException Class

An AdsServerException is thrown on communication errors in the AdsServer [▶ 1121] class.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException
      TwinCAT.Ads.Server.LoopbackNotRegisteredException [▶ 1188]
      TwinCAT.Ads.Server.ServerNotConnectedException [▶ 1202]

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class AdsServerException : Exception
```

The AdsServerException type exposes the following members.

Constructors

	Name	Description
	AdsServerException [▶ 1183]	Initializes a new instance of the AdsServerException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 1184]	Gets or sets the error code.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.1 AdsServerException Constructor

Initializes a new instance of the AdsServerException [▶ 1181] class.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AdsServerException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.2 AdsServerException Properties

The AdsServerException [▶ 1181] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 1184]	Gets or sets the error code.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)

	Name	Description
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.2.1 AdsServerException.ErrorCode Property

Gets or sets the error code.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsServerErrorCode ErrorCode { get; set; }
```

Property Value

Type: AmsServerErrorCode [▶ 1548]
The error code.

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.3 AdsServerException Methods

The AdsServerException [▶ 1181] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.3.1 AdsServerException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.2.4 AdsServerException Events

The AdsServerException [▶ 1181] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

AdsServerException Class [▶ 1181]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.3 ErrorEventArgs Class

This class implements the event arguments passed by the TcAdsServerExEvent.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.Server.ErrorEventArgs
Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ErrorEventArgs : EventArgs
```

The ErrorEventArgs type exposes the following members.

Properties

	Name	Description
	Exception [▶ 1187]	Returns the exception that caused the event.
	Message [▶ 1188]	Returns the exception message.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.3.1 ErrorEventArgs Properties

The ErrorEventArgs [▶ 1186] type exposes the following members.

Properties

	Name	Description
	Exception [▶ 1187]	Returns the exception that caused the event.
	Message [▶ 1188]	Returns the exception message.

Reference

ErrorEventArgs Class [▶ 1186]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.3.1.1 ErrorEventArgs.Exception Property

Returns the exception that caused the event.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Exception Exception { get; }
```

Return Value

Type: Exception
The Exception that caused the event.

Reference

ErrorEventArgs Class [▶ 1186]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.3.1.2 ErrorEventArgs.Message Property

Returns the exception message.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Message { get; }
```

Return Value

Type: String
The Exception message.

Reference

ErrorEventArgs Class [▶ 1186]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.3.2 ErrorEventArgs Methods

The ErrorEventArgs [▶ 1186] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ErrorEventArgs Class [▶ 1186]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.4 LoopbackNotRegisteredException Class

The Tcp Loopback client is not registered. Implements the AdsServerException [▶ 1181]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException [▶ 1181]
      TwinCAT.Ads.Server.LoopbackNotRegisteredException

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class LoopbackNotRegisteredException : AdsServerException
```

The LoopbackNotRegisteredException type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 1184]	Gets or sets the error code. (Inherited from AdsServerException [▶ 1181].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException [▶ 1181].)

	Name	Description
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

TwinCAT.Ads.Server.AdsServerException [▶ 1181]

6.4.4.1 LoopbackNotRegisteredException Properties

The LoopbackNotRegisteredException [▶ 1188] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 1184]	Gets or sets the error code. (Inherited from AdsServerException [▶ 1181].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

LoopbackNotRegisteredException Class [▶ 1188]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.4.2 LoopbackNotRegisteredException Methods

The LoopbackNotRegisteredException [▶ 1188] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException [▶ 1181].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

LoopbackNotRegisteredException Class [▶ 1188]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.4.3 LoopbackNotRegisteredException Events

The LoopbackNotRegisteredException [▶ 1188] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

LoopbackNotRegisteredException Class [▶ 1188]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5 NotificationDataSample Class

This class implements an ADS Notification Sample. It contains the notification handle and the variable data.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Server.NotificationDataSample
Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class NotificationDataSample
```

The NotificationDataSample type exposes the following members.

Constructors

	Name	Description
	NotificationDataSample [▶ 1192]	

Properties

	Name	Description
	NotificationHandle [▶ 1193]	Gets the notification handle.
	SampleData [▶ 1194]	Gets the Sample Data.
	SampleSize [▶ 1194]	Gets the Sample Size.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.1 NotificationDataSample Constructor

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NotificationDataSample(
    uint handle,
    ReadOnlyMemory data
)
```

Parameters

handle	Type: System.UInt32
data	Type: ReadOnlyMemory

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.2 NotificationDataSample Properties

The NotificationDataSample [▶ 1191] type exposes the following members.

Properties

	Name	Description
	NotificationHandle [▶ 1193]	Gets the notification handle.
	SampleData [▶ 1194]	Gets the Sample Data.
	SampleSize [▶ 1194]	Gets the Sample Size.

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.2.1 NotificationDataSample.NotificationHandle Property

Gets the notification handle.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint NotificationHandle { get; }
```

Property Value

Type: UInt32

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.2.2 NotificationDataSample.SampleData Property

Gets the Sample Data.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemory SampleData { get; }
```

Property Value

Type: ReadOnlyMemory

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.2.3 NotificationDataSample.SampleSize Property

Gets the Sample Size.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int SampleSize { get; }
```

Property Value

Type: Int32

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.5.3 NotificationDataSample Methods

The NotificationDataSample [▶ 1191] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

NotificationDataSample Class [▶ 1191]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6 NotificationSamplesStamp Class

This class implements an ADS Stamp Header containing multiple ADS Notification Samples (TcAdsStampHeader)

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.Server.NotificationSamplesStamp
Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class NotificationSamplesStamp
```

The NotificationSamplesStamp type exposes the following members.

Constructors

	Name	Description
	NotificationSamplesStamp [▶ 1196]	Initializes a new instance of the NotificationSamplesStamp class.

Properties

	Name	Description
	NotificationSamples [▶ 1197]	Get the array of notification samples contained in this header.
	NumSamples [▶ 1197]	Get the number of notification samples contained in this header.
	TimeStamp [▶ 1198]	Get or set the time stamp of this header.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.1 NotificationSamplesStamp Constructor

Initializes a new instance of the NotificationSamplesStamp [▶ 1195] class.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NotificationSamplesStamp(
    long timestamp,
    NotificationDataSample[] samples
)
```

Parameters

timestamp	Type: System.Int64 The timestamp.
samples	Type: .TwinCAT.Ads.Server.NotificationDataSample [▶ 1191]. The notification samples.

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.2 NotificationSamplesStamp Properties

The NotificationSamplesStamp [▶ 1195] type exposes the following members.

Properties

	Name	Description
	NotificationSamples [▶ 1197]	Get the array of notification samples contained in this header.
	NumSamples [▶ 1197]	Get the number of notification samples contained in this header.

	Name	Description
	TimeStamp [▶ 1198]	Get or set the time stamp of this header.

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.2.1 NotificationSamplesStamp.NotificationSamples Property

Get the array of notification samples contained in this header.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NotificationDataSample[] NotificationSamples { get; }
```

Property Value

Type: .NotificationDataSample [▶ 1191].

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.2.2 NotificationSamplesStamp.NumSamples Property

Get the number of notification samples contained in this header.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int NumSamples { get; }
```

Property Value

Type: Int32

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.2.3 NotificationSamplesStamp.TimeStamp Property

Get or set the time stamp of this header.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public long TimeStamp { get; }
```

Property Value

Type: Int64

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.6.3 NotificationSamplesStamp Methods

The NotificationSamplesStamp [▶ 1195] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

NotificationSamplesStamp Class [▶ 1195]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.7 ServerConnectionState Enumeration

The Server Connection State

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum ServerConnectionState
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized / Disconnect
	Disconnected	1	Disconnected State.
	Connected	2	Connected state.

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.8 ServerConnectionStateChangedEventArgs Class

Class ConnectionStateChangedEventArgs (Server Connections)

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.Server.ServerConnectionStateChangedEventArgs

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ServerConnectionStateChangedEventArgs : EventArgs
```

The ServerConnectionStateChangedEventArgs type exposes the following members.

Constructors

	Name	Description
	ServerConnectionStateChangedEventArgs [▶ 1200]	Initializes a new instance of the ServerConnectionStateChangedEventArgs class.

Properties

	Name	Description
	Address [▶ 1201]	The Server address
	State [▶ 1201]	The Connection state

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

System.EventArgs

6.4.8.1 ServerConnectionStateChangedEventArgs Constructor

Initializes a new instance of the ServerConnectionStateChangedEventArgs [▶ 1199] class.

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ServerConnectionStateChangedEventArgs(
    AmsAddress address,
    ServerConnectionState state
)
```

Parameters

address	Type: TwinCAT.Ads.AmsAddress [▶ 648] The address.
state	Type: TwinCAT.Ads.Server.ServerConnectionState [▶ 1198] The state.

Reference

ServerConnectionStateChangedEventArgs Class [▶ 1199]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.8.2 ServerConnectionStateChangedEventArgs Properties

The ServerConnectionStateChangedEventArgs [▶ 1199] type exposes the following members.

Properties

	Name	Description
	Address [▶ 1201]	The Server address
	State [▶ 1201]	The Connection state

Reference

ServerConnectionStateChangedEventArgs Class [▶ 1199]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.8.2.1 ServerConnectionStateChangedEventArgs.Address Property

The Server address

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress Address { get; }
```

Property Value

Type: AmsAddress [▶ 648]

Reference

ServerConnectionStateChangedEventArgs Class [▶ 1199]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.8.2.2 ServerConnectionStateChangedEventArgs.State Property

The Connection state

Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ServerConnectionState State { get; }
```

Property Value

Type: ServerConnectionState [▶ 1198]

Reference

ServerConnectionStateChangedEventArgs Class [▶ 1199]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.8.3 ServerConnectionStateChangedEventArgs Methods

The ServerConnectionStateChangedEventArgs [▶ 1199] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ServerConnectionStateChangedEventArgs Class [▶ 1199]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.9 ServerNotConnectedException Class

The AdsServer is not connected. Implements the AdsServerException [▶ 1181]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.Server.AdsServerException [▶ 1181]
      TwinCAT.Ads.Server.ServerNotConnectedException
Namespace: TwinCAT.Ads.Server [▶ 1121]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ServerNotConnectedException : AdsServerException
```

The ServerNotConnectedException type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	ErrorCode [▶ 1184]	Gets or sets the error code. (Inherited from AdsServerException [▶ 1181].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException [▶ 1181].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.Server Namespace [▶ 1121]

TwinCAT.Ads.Server.AdsServerException [▶ 1181]

6.4.9.1 ServerNotConnectedException Properties

The ServerNotConnectedException [▶ 1202] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)

	Name	Description
	ErrorCode [▶ 1184]	Gets or sets the error code. (Inherited from AdsServerException [▶ 1181].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

ServerNotConnectedException Class [▶ 1202]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.9.2 ServerNotConnectedException Methods

The ServerNotConnectedException [▶ 1202] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1185]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from AdsServerException [▶ 1181].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

ServerNotConnectedException Class [▶ 1202]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.4.9.3 ServerNotConnectedException Events

The ServerNotConnectedException [▶ 1202] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

ServerNotConnectedException Class [▶ 1202]

TwinCAT.Ads.Server Namespace [▶ 1121]

6.5 TwinCAT.Ads.SumCommand Namespace

ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. ADS Sum-Commands offer to read/write with one single ADS call multiple variables which are not structured within a linear memory, effectively reducing roundtrips.

Classes

	Class	Description
	ResultSumCommand [▶ 1210]	Result class for an asynchronous TwinCAT.Ads.SumCommand:
	ResultSumHandles [▶ 1212]	Result class for an asynchronous SumCreateHandles [▶ 1222].
	ResultSumHandles2 [▶ 1214]	Result class for an asynchronous SumCreateHandles [▶ 1222].
	ResultSumReadRaw [▶ 1218]	Result class for an asynchronous SumRead.
	ResultSumValues [▶ 1220]	Result class for an asynchronous SumSymbolRead [▶ 1242].
	SumCreateHandles [▶ 1222]	SumCommandBase for getting variable handles by a set of InstancePaths
	SumHandleRead [▶ 1228]	Read (primitive, Any) values by Handle SumCommandBase.
	SumHandleWrite [▶ 1232]	Write any (primitive) values by Handle SumCommandBase.
	SumReleaseHandles [▶ 1237]	Release Handles SumCommandBase.

	Class	Description
	SumSymbolRead [▶ 1242]	Symbolic ADS Sum read access
	SumSymbolWrite [▶ 1248]	Class for ADS Sum symbolic Write Access.

Interfaces

	Interface	Description
	ISumCommand [▶ 1206]	Interface for SumCommands (Combined commands)

6.5.1 ISumCommand Interface

Interface for SumCommands (Combined commands)

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISumCommand
```

The ISumCommand type exposes the following members.

Properties

	Name	Description
	Executed [▶ 1207]	Gets a value indicating whether this ISumCommand was already executed.
	Failed [▶ 1208]	Gets a value indicating whether this ISumCommand failed.
	Result [▶ 1208]	Gets the AdsErrorCode [▶ 575] of the main SumCommandBase ADS Request
	SubResults [▶ 1209]	Gets the sub results of the single Sub Requests.
	Succeeded [▶ 1209]	Gets a value indicating whether this ISumCommand is succeeded.

Remarks

ADS offers powerful and fast communication to exchange any kind of information. It's possible to read single variables or complete arrays and structures with each one single ADS-API call. The ADS Sum Command command offers to read with one singale ADS call multiple variables which are not structured within a linear memory. As a result the ADS caller application (like scada Systems etc.) can extremly speed up cyclic polling : Sample :

- Until now : Polling 4000 single variables which are not in a linear area (like array / structure / fixed PLC address) would cause 4000 single Ads-ReadReq with each 1-2 ms protocol time. As a result the scanning of these variables take 4000ms-8000ms.

- New Ads-Command allows to read multiple variables with one single ADS-ReadReq : 4000 single variables are handled with e.g. 8 single Ads-ReadReq (each call requesting 500 variables) with each 1-2 ms protocol time. As a result the scanning of these variables take just few 10ms.

REQUIREMENTS AND IMPORTANT LIMITATIONS: Note that ADS is just a transport layer, but there could be important side effects. So read this requirements and take care on limitations:

- **Version of target ADS Device** - - ADS itself is just the transport layer, but the requested ADS device has to support the ADS-Command.
- **Bytes length of requested data** - - Requesting a large list of values from variables is fine, but the requested data of the Ads-response (the data-byte-length) have to pass the AMS Router (size by default a 2048kb) So the caller has to limit the requested variables based on calculation of requested data-byte-length.
- **Number of Sub-ADS calls : Highly recommended to max. 500 !** - - If the PLC is processing one ADS request, it will completely work on this single ADS request BEFORE starting neyt PLC cycle. As a result one single ADS request with 200.000 sub-Ads-requests would cause that PLC would collect and copy 200.000 variables into one single ADS response, before starting next PLC. So this large number of ads-sub-commands will jitter the PLC execution ! **We highly recommend to not request more than 500 Ads-Sub commands**

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1 ISumCommand Properties

The ISumCommand [▶ 1206] type exposes the following members.

Properties

	Name	Description
	Executed [▶ 1207]	Gets a value indicating whether this ISumCommand [▶ 1206] was already executed.
	Failed [▶ 1208]	Gets a value indicating whether this ISumCommand [▶ 1206] failed.
	Result [▶ 1208]	Gets the AdsErrorCode [▶ 575] of the main SumCommandBase ADS Request
	SubResults [▶ 1209]	Gets the sub results of the single Sub Requests.
	Succeeded [▶ 1209]	Gets a value indicating whether this ISumCommand [▶ 1206] is succeeded.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1.1 ISumCommand.Executed Property

Gets a value indicating whether this ISumCommand [▶ 1206] was already executed.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Executed { get; }
```

Property Value

Type: Boolean
true if executed; otherwise, false.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1.2 ISumCommand.Failed Property

Gets a value indicating whether this ISumCommand [▶ 1206] failed.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Failed { get; }
```

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1.3 ISumCommand.Result Property

Gets the AdsErrorCode [▶ 575] of the main SumCommandBase ADS Request

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode Result { get; }
```

Property Value

Type: AdsErrorCode [▶ 575]
The result.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1.4 ISumCommand.SubResults Property

Gets the sub results of the single Sub Requests.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsErrorCode[] SubResults { get; }
```

Property Value

Type: .AdsErrorCode [▶ 575].
The sub results.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.1.1.5 ISumCommand.Succeeded Property

Gets a value indicating whether this ISumCommand [▶ 1206] is succeeded.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Succeeded { get; }
```

Property Value

Type: Boolean
true if succeeded; otherwise, false.

Reference

ISumCommand Interface [▶ 1206]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.2 ResultSumCommand Class

Result class for an asynchronous TwinCAT.Ads.SumCommand [▶ 1205]:

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.SumCommand.ResultSumCommand
      TwinCAT.Ads.SumCommand.ResultSumHandles [▶ 1212]
      TwinCAT.Ads.SumCommand.ResultSumReadRaw [▶ 1218]
      TwinCAT.Ads.SumCommand.ResultSumValues [▶ 1220]

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSumCommand : ResultAds
```

The ResultSumCommand type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.2.1 ResultSumCommand Properties

The ResultSumCommand [▶ 1210] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands.
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultSumCommand Class [▶ 1210]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.2.1.1 ResultSumCommand.SubErrors Property

Gets the Error codes [▶ 575] for the single SumCommands.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode[] SubErrors { get; protected set; }
```

Property Value

Type: .AdsErrorCode [▶ 575].

Reference

ResultSumCommand Class [▶ 1210]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.2.2 ResultSumCommand Methods

The ResultSumCommand [▶ 1210] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSumCommand Class [▶ 1210]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.3 ResultSumHandles Class

Result class for an asynchronous SumCreateHandles [▶ 1222].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [▶ 1210]
      TwinCAT.Ads.SumCommand.ResultSumHandles
        TwinCAT.Ads.SumCommand.ResultSumHandles2 [▶ 1214]

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSumHandles : ResultSumCommand
```

The ResultSumHandles type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handles [▶ 1213]	The registered Symbol handles.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.3.1 ResultSumHandles Properties

The ResultSumHandles [▶ 1212] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handles [▶ 1213]	The registered Symbol handles.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultSumHandles Class [▶ 1212]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.3.1.1 ResultSumHandles.Handles Property

The registered Symbol handles.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint[] Handles { get; }
```

Property Value

Type: .UInt32.

Reference

ResultSumHandles Class [▶ 1212]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.3.2 ResultSumHandles Methods

The ResultSumHandles [▶ 1212] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSumHandles Class [▶ 1212]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.4 ResultSumHandles2 Class

Result class for an asynchronous SumCreateHandles [▶ 1222].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [▶ 1210]
      TwinCAT.Ads.SumCommand.ResultSumHandles [▶ 1212]
        TwinCAT.Ads.SumCommand.ResultSumHandles2

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSumHandles2 : ResultSumHandles
```

The ResultSumHandles2 type exposes the following members.

Constructors

	Name	Description
	ResultSumHandles2 [▶ 1216]	Initializes a new instance of the ResultSumHandles2 class.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handles [▶ 1213]	The registered Symbol handles. (Inherited from ResultSumHandles [▶ 1212].)
	InstancePaths [▶ 1217]	The symbol/instance paths belonging to the handles and the subErrors.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.4.1 ResultSumHandles2 Constructor

Initializes a new instance of the ResultSumHandles2 [▶ 1214] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultSumHandles2(
    AdsErrorCode complete,
    AdsErrorCode[] subErrors,
    uint[] handles,
    string[] instancePaths
)
```

Parameters

complete	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The overall ADS error codes.
subErrors	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575]. The single ADS SubErrors.
handles	Type: .System.UInt32. The registered handles.
instancePaths	Type: .System.String. The instance paths.

Reference

ResultSumHandles2 Class [▶ 1214]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.4.2 ResultSumHandles2 Properties

The ResultSumHandles2 [▶ 1214] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Handles [▶ 1213]	The registered Symbol handles. (Inherited from ResultSumHandles [▶ 1212].)
	InstancePaths [▶ 1217]	The symbol/instance paths belonging to the handles and the subErrors.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultSumHandles2 Class [▶ 1214]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.4.2.1 ResultSumHandles2.InstancePaths Property

The symbol/instance paths belonging to the handles and the subErrors.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] InstancePaths { get; }
```

Property Value

Type: .String.

Reference

ResultSumHandles2 Class [▶ 1214]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.4.3 ResultSumHandles2 Methods

The ResultSumHandles2 [▶ 1214] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSumHandles2 Class [▶ 1214]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.5 ResultSumReadRaw Class

Result class for an asynchronous SumRead.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [▶ 1210]
      TwinCAT.Ads.SumCommand.ResultSumReadRaw

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSumReadRaw : ResultSumCommand
```

The ResultSumReadRaw type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBlocks [▶ 1219]	List of the read raw memory blocks read.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.5.1 ResultSumReadRaw Properties

The ResultSumReadRaw [▶ 1218] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	ReadBlocks [▶ 1219]	List of the read raw memory blocks read.
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)

Reference

ResultSumReadRaw Class [▶ 1218]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.5.1.1 ResultSumReadRaw.ReadBlocks Property

List of the read raw memory blocks read.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IList<byte[]> ReadBlocks { get; }
```

Property Value

Type: IList..Byte..

Reference

ResultSumReadRaw Class [▶ 1218]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.5.2 ResultSumReadRaw Methods

The ResultSumReadRaw [▶ 1218] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSumReadRaw Class [▶ 1218]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.6 ResultSumValues Class

Result class for an asynchronous SumSymbolRead [▶ 1242].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.SumCommand.ResultSumCommand [▶ 1210]
      TwinCAT.Ads.SumCommand.ResultSumValues

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSumValues : ResultSumCommand
```

The ResultSumValues type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)

	Name	Description
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Values [▶ 1222]	The Read data as marshalled values

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.6.1 ResultSumValues Properties

The ResultSumValues [▶ 1220] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	SubErrors [▶ 1211]	Gets the Error codes [▶ 575] for the single SumCommands. (Inherited from ResultSumCommand [▶ 1210].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Values [▶ 1222]	The Read data as marshalled values

Reference

ResultSumValues Class [▶ 1220]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.6.1.1 ResultSumValues.Values Property

The Read data as marshalled values

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object[] Values { get; }
```

Property Value

Type: .Object.

Reference

ResultSumValues Class [▶ 1220]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.6.2 ResultSumValues Methods

The ResultSumValues [▶ 1220] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSumValues Class [▶ 1220]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.7 SumCreateHandles Class

SumCommandBase for getting variable handles by a set of InstancePaths

Inheritance Hierarchy

System.Object
  SumCommandWrapper.SumReadWrite.
    TwinCAT.Ads.SumCommand.SumCreateHandles
Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumCreateHandles : SumCommandWrapper<SumReadWrite>
```

The SumCreateHandles type exposes the following members.

Constructors

	Name	Description
	SumCreateHandles(IAdsConnection, IList.String.) [▶ 1225]	Initializes a new instance of the SumCreateHandles class.
	SumCreateHandles(IAdsConnection, .String.) [▶ 1225]	Initializes a new instance of the SumCreateHandles class.

Methods

	Name	Description
	CreateHandles [▶ 1227]	Creates the ADS handles.
	CreateHandlesAsync [▶ 1227]	Create handles asynchronously.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Examples

Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
```

```
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.Proje
ctName"};
    SumCreateHandles createHandlesCommand = new SumCreateHandles(client,instancePathList);

    var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);

    if (resultCreateHandles.Succeeded)
    {
        uint[] handles = resultCreateHandles.Handles;
        Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
        SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
        var resultRead = await readCommand.ReadAsync(cancel);

        if (resultRead.Succeeded)
        {
        object[] readValues = resultRead.Values;

        for (int i = 0; i < instancePathList.Length; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues
[i].ToString(), valueTypes[i].Name);
        }

        // Sum Command Write
        SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        await writeCommand.WriteAsync(writeValues,cancel);
        }

        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.SumHandleRead [▶ 1228]

TwinCAT.Ads.SumCommand.SumHandleWrite [▶ 1232]

6.5.7.1 SumCreateHandles Constructor

Overload List

	Name	Description
	SumCreateHandles(IAdsConnection, IList.String.) [▶ 1225]	Initializes a new instance of the SumCreateHandles [▶ 1222] class.
	SumCreateHandles(IAdsConnection, .String.) [▶ 1225]	Initializes a new instance of the SumCreateHandles [▶ 1222] class.

Reference

SumCreateHandles Class [▶ 1222]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.7.1.1 SumCreateHandles Constructor (IAdsConnection, IList.String.)

Initializes a new instance of the SumCreateHandles [▶ 1222] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SumCreateHandles(
    IAdsConnection connection,
    IList<string> instancePaths
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePaths	Type: System.Collections.Generic.IList.String. The instance paths.

Reference

SumCreateHandles Class [▶ 1222]

SumCreateHandles Overload [▶ 1225]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.7.1.2 SumCreateHandles Constructor (IAdsConnection, .String.)

Initializes a new instance of the SumCreateHandles [▶ 1222] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SumCreateHandles(
    IAdsConnection connection,
    string[] instancePaths
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
instancePaths	Type: .System.String. The instance paths.

Reference

SumCreateHandles Class [▶ 1222]

SumCreateHandles Overload [▶ 1225]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.7.2 SumCreateHandles Methods

The SumCreateHandles [▶ 1222] type exposes the following members.

Methods

	Name	Description
	CreateHandles [▶ 1227]	Creates the ADS handles.
	CreateHandlesAsync [▶ 1227]	Create handles asynchronously.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SumCreateHandles Class [▶ 1222]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.7.2.1 SumCreateHandles.CreateHandles Method

Creates the ADS handles.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint[] CreateHandles()
```

Return Value

Type: .UInt32.
System.UInt32[].

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumGetHandles failed!
AdsSumCommandException [▶ 632]	SumGetHandlesCommand failed!

Reference

SumCreateHandles Class [▶ 1222]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumCreateHandles.TryCreateHandles(.String.., .UInt32.., .AdsErrorCode..)

SumCreateHandles.CreateHandlesAsync(CancellationToken) [▶ 1227]

6.5.7.2.2 SumCreateHandles.CreateHandlesAsync Method

Create handles asynchronously.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumHandles> CreateHandlesAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultSumHandles [▶ 1212].
An asynchronous task that represents the 'ReadWriteRaw' operation and returns a ResultSumHandles [▶ 1212]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumCreateHandles Class [▶ 1222]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumCreateHandles.CreateHandles. [▶ 1227]

SumCreateHandles.TryCreateHandles(.String.., .UInt32.., .AdsErrorCode..)

6.5.8 SumHandleRead Class

Read (primitive, Any) values by Handle SumCommandBase.

Inheritance Hierarchy

System.Object
  SumCommandBase
    SumRead
      TwinCAT.Ads.SumCommand.SumHandleRead
Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumHandleRead : SumRead
```

The SumHandleRead type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Read [▶ 1230]	Reads the values.
	ReadAsync [▶ 1231]	Read the values asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryRead [▶ 1232]	Tries to read the values of the .

Remarks

This is an ADS Sum Command to access values by handle information. It is always used in combination with and . By design (and in contrast to the symbolic access in SumSymbolRead [▶ 1242], SumSymbolWrite [▶ 1248]) this access method can act only with ADS ANY Type (Primitive) values (disadvantage). The Advantage is, that no symbolic information must be loaded before accessing the values, see samples:

Examples

Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.Proje
ctName"};
    SumCreateHandles createHandlesCommand = new SumCreateHandles(client,instancePathList);

    var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);

    if (resultCreateHandles.Succeeded)
    {
        uint[] handles = resultCreateHandles.Handles;
        Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
        SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
        var resultRead = await readCommand.ReadAsync(cancel);

        if (resultRead.Succeeded)
        {
        object[] readValues = resultRead.Values;

        for (int i = 0; i < instancePathList.Length; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues
[i].ToString(), valueTypes[i].Name);
        }

        // Sum Command Write
        SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        await writeCommand.WriteAsync(writeValues,cancel);
        }

        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.SumCreateHandles [▶ 1222]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.SumHandleWrite [▶ 1232]

6.5.8.1 SumHandleRead Methods

The SumHandleRead [▶ 1228] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Read [▶ 1230]	Reads the values.
	ReadAsync [▶ 1231]	Read the values asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryRead [▶ 1232]	Tries to read the values of the .

Reference

SumHandleRead Class [▶ 1228]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.8.1.1 SumHandleRead.Read Method

Reads the values.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object[] Read()
```

Return Value

Type: .Object.
System.Object[].

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumAnyReadByHandleCommand failed!

Reference

SumHandleRead Class [▶ 1228]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumHandleRead.TryRead(.Object.., .AdsErrorCode..) [▶ 1232]

SumHandleRead.ReadAsync(CancellationToken) [▶ 1231]

6.5.8.1.2 SumHandleRead.ReadAsync Method

Read the values asynchronously.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumValues> ReadAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultSumValues [▶ 1220].
An asynchronous task that represents the 'ReadSymbols' operation and returns a ResultSumValues [▶ 1220]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumHandleRead Class [▶ 1228]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumHandleRead.Read. [▶ 1230]

SumHandleRead.TryRead(.Object.., .AdsErrorCode..) [▶ 1232]

6.5.8.1.3 SumHandleRead.TryRead Method

Tries to read the values of the .

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    out Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values	Type: .System.Object.. The values.
returnCodes	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575].. The return codes.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

SumHandleRead Class [▶ 1228]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.SumHandleRead [▶ 1228]

SumHandleRead.Read. [▶ 1230]

SumHandleRead.ReadAsync(CancellationToken) [▶ 1231]

6.5.9 SumHandleWrite Class

Write any (primitive) values by Handle SumCommandBase.

Inheritance Hierarchy

System.Object
  SumCommandBase
    SumWrite
      TwinCAT.Ads.SumCommand.SumHandleWrite

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumHandleWrite : SumWrite
```

The SumHandleWrite type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryWrite [▶ 1235]	Tries to write the values.
	Write [▶ 1235]	Writes the values to the Symbols.
	WriteAsync [▶ 1236]	Write the values asynchronously.

Remarks

This is an ADS Sum Command to access values by handle information. It is always used in combination with and . By design (and in contrast to the symbolic access in SumSymbolRead [▶ 1242], SumSymbolWrite [▶ 1248]) this access method can act only with ADS ANY Type (Primitive) values (disadvantage). The Advantage is, that no symbolic information must be loaded before accessing the values, see samples:

Examples

Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.Proje
ctName"};
    SumCreateHandles createHandlesCommand = new SumCreateHandles(client,instancePathList);

    var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);

    if (resultCreateHandles.Succeeded)
    {
        uint[] handles = resultCreateHandles.Handles;
        Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
        SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
        var resultRead = await readCommand.ReadAsync(cancel);
```

```
        if (resultRead.Succeeded)
        {
        object[] readValues = resultRead.Values;

        for (int i = 0; i < instancePathList.Length; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues
[i].ToString(), valueTypes[i].Name);
        }

        // Sum Command Write
        SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        await writeCommand.WriteAsync(writeValues,cancel);
        }

        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.SumCreateHandles [▶ 1222]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.SumReleaseHandles [▶ 1237]

TwinCAT.Ads.SumCommand.SumHandleRead [▶ 1228]

6.5.9.1 SumHandleWrite Methods

The SumHandleWrite [▶ 1232] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryWrite [▶ 1235]	Tries to write the values.

	Name	Description
	Write [▶ 1235]	Writes the values to the Symbols.
	WriteAsync [▶ 1236]	Write the values asynchronously.

Reference

SumHandleWrite Class [▶ 1232]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.9.1.1 SumHandleWrite.TryWrite Method

Tries to write the values.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values	Type: .System.Object. The values (ANY/Primitive types only).
returnCodes	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575].. The return codes.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

SumHandleWrite Class [▶ 1232]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumHandleWrite.Write(.Object.) [▶ 1235]

SumHandleWrite.WriteAsync(.Object., CancellationToken) [▶ 1236]

6.5.9.1.2 SumHandleWrite.Write Method

Writes the values to the Symbols.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    Object[] values
)
```

Parameters

values	Type: .System.Object. The Values (Any primitive types only):

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumAnyWriteByHandleCommand failed!

Reference

SumHandleWrite Class [▶ 1232]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumHandleWrite.TryWrite(.Object., .AdsErrorCode..) [▶ 1235]

SumHandleWrite.WriteAsync(.Object., CancellationToken) [▶ 1236]

6.5.9.1.3 SumHandleWrite.WriteAsync Method

Write the values asynchronously.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumCommand> WriteAsync(
    Object[] values,
    CancellationToken cancel
)
```

Parameters

values	Type: .System.Object. The values.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultSumCommand [▶ 1210].
An asynchronous task that represents the 'ReadWriteRaw' operation and returns a ResultSumCommand [▶ 1210]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumHandleWrite Class [▶ 1232]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumHandleWrite.Write(.Object.) [▶ 1235]

SumHandleWrite.TryWrite(.Object., .AdsErrorCode..) [▶ 1235]

6.5.10 SumReleaseHandles Class

Release Handles SumCommandBase.

Inheritance Hierarchy

System.Object
  SumCommandWrapper.SumWrite.
    TwinCAT.Ads.SumCommand.SumReleaseHandles
Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumReleaseHandles : SumCommandWrapper<SumWrite>
```

The SumReleaseHandles type exposes the following members.

Constructors

	Name	Description
	SumReleaseHandles [▶ 1239]	Initializes a new instance of the SumReleaseHandles class.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ReleaseHandles [▶ 1240]	Releases the handles.
	ReleaseHandlesAsync [▶ 1240]	Releases the handles asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

	Name	Description
	TryReleaseHandles [▶ 1241]	Tries to Release the Handles

Remarks

Releases the specified ADS handles. Usually used in conjunction with the SumCreateHandles [▶ 1222] and the SumHandleRead [▶ 1228] / SumHandleWrite [▶ 1232] commands.

Examples

Usage of Sum commands with handles (CreateHandles, Read, Write, ReleaseHandles)

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    string[] instancePathList = {"GVL.bVar", "GVL.iCount", "TwinCAT_SystemInfoVarList._AppInfo.Proje
ctName"};
    SumCreateHandles createHandlesCommand = new SumCreateHandles(client,instancePathList);

    var resultCreateHandles = await createHandlesCommand.CreateHandlesAsync(cancel);

    if (resultCreateHandles.Succeeded)
    {
        uint[] handles = resultCreateHandles.Handles;
        Type[] valueTypes = new Type[] { typeof(bool), typeof(short), typeof(string) };
        SumHandleRead readCommand = new SumHandleRead(client, handles, valueTypes);
        var resultRead = await readCommand.ReadAsync(cancel);

        if (resultRead.Succeeded)
        {
        object[] readValues = resultRead.Values;

        for (int i = 0; i < instancePathList.Length; i++)
        {
            Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", instancePathList[i], readValues
[i].ToString(), valueTypes[i].Name);
        }

        // Sum Command Write
        SumHandleWrite writeCommand = new SumHandleWrite(client, handles, valueTypes);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        await writeCommand.WriteAsync(writeValues,cancel);
        }

        SumReleaseHandles releaseCommand = new SumReleaseHandles(client, handles);
        await releaseCommand.ReleaseHandlesAsync(cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumCreateHandles [▶ 1222]

TwinCAT.Ads.SumCommand.SumHandleRead [▶ 1228]

TwinCAT.Ads.SumCommand.SumHandleWrite [▶ 1232]

6.5.10.1 SumReleaseHandles Constructor

Initializes a new instance of the SumReleaseHandles [▶ 1237] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SumReleaseHandles(
    IAdsConnection connection,
    uint[] serverHandles
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The connection.
serverHandles	Type: .System.UInt32. The handles.

Reference

SumReleaseHandles Class [▶ 1237]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.10.2 SumReleaseHandles Methods

The SumReleaseHandles [▶ 1237] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ReleaseHandles [▶ 1240]	Releases the handles.
	ReleaseHandlesAsync [▶ 1240]	Releases the handles asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryReleaseHandles [▶ 1241]	Tries to Release the Handles

Reference

SumReleaseHandles Class [▶ 1237]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.10.2.1 SumReleaseHandles.ReleaseHandles Method

Releases the handles.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void ReleaseHandles()
```

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumReleaseHandlesCommand failed!

Reference

SumReleaseHandles Class [▶ 1237]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumReleaseHandles.TryReleaseHandles(.AdsErrorCode..) [▶ 1241]

SumReleaseHandles.ReleaseHandlesAsync(CancellationToken) [▶ 1240]

6.5.10.2.2 SumReleaseHandles.ReleaseHandlesAsync Method

Releases the handles asynchronously.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumCommand> ReleaseHandlesAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultSumCommand [▶ 1210].
An asynchronous task that represents the 'ReleaseHandles' operation and returns a ResultSumCommand [▶ 1210]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumReleaseHandles Class [▶ 1237]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.10.2.3 SumReleaseHandles.TryReleaseHandles Method

Tries to Release the Handles

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryReleaseHandles(
    out AdsErrorCode[] returnCodes
)
```

Parameters

returnCodes	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575].. The return codes.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Reference

SumReleaseHandles Class [▶ 1237]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumReleaseHandles.ReleaseHandles. [▶ 1240]

SumReleaseHandles.ReleaseHandlesAsync(CancellationToken) [▶ 1240]

6.5.11 SumSymbolRead Class

Symbolic ADS Sum read access

Inheritance Hierarchy

System.Object
  SumCommandWrapper.SumRead.
    SumSymbolCommand.SumRead.
      TwinCAT.Ads.SumCommand.SumSymbolRead
Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumSymbolRead : SumSymbolCommand<SumRead>
```

The SumSymbolRead type exposes the following members.

Constructors

	Name	Description
	SumSymbolRead [▶ 1244]	Initializes a new instance of the SumSymbolRead class.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Read [▶ 1245]	Reads the Values.
	ReadAsync [▶ 1246]	Reads Symbol values as an asynchronous operation.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryRead [▶ 1247]	Reads the specified symbols.

Remarks

The SumSymbolRead implements symbolic read access with automatic (dynamic) value marshalling. The advantage of the symbolic access is (in contrast to the handle access classes SumHandleRead [▶ 1228],SumHandleWrite [▶ 1232]) that all type information is available when using this ADS Sum Command. The disadvantage is, that the Symbolic information must be loaded beforehand, see examples.

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsSession

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsSession session = new AdsSession(address))
    {
    // Connect to the device target.
    AdsConnection connection = (AdsConnection)session.Connect();

    // Load symbolic information
    var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);
    resultSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;

    ISymbol bVar1 = symbols["GVL.bVar1"];
    ISymbol bVar2 = symbols["GVL.bVar2"];
    ISymbol projectName = symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];

    SymbolCollection coll = new SymbolCollection() {bVar1, bVar2, projectName};

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(connection,coll);
    var resultReadValues = await readCommand.ReadAsync(cancel);

    if (resultReadValues.Succeeded)
    {
        object[] values = resultReadValues.Values;

        for (int i = 0; i < coll.Count; i++)
        {
        Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", coll[i].InstancePath, values[i].ToS
tring(), values[i].GetType().Name);
        }
    }
    // Sum Command Write

    SumSymbolWrite writeCommand = new SumSymbolWrite(connection,coll);
    object[] writeValues = new object[] {true, (short) 42, "MyNewProjectName"};

    var resultWrite = await writeCommand.WriteAsync(writeValues,cancel);
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsClient

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();
```

```
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    // Load symbolic information
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    var resultReadSymbols = await loader.GetSymbolsAsync(cancel);

    resultReadSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> allSymbols = resultReadSymbols.Symbols;

    ISymbol bVar1 = allSymbols["GVL.bVar1"];
    ISymbol bVar2 = allSymbols["GVL.iCount"];
    ISymbol projectName = allSymbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];

    SymbolCollection symbols = new SymbolCollection() {bVar1, bVar2, projectName};

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(client,symbols);
    var resultSumRead = await readCommand.ReadAsync(cancel);

    if (resultSumRead.Succeeded)
    {
        object[] values = resultSumRead.Values;

        for (int i = 0; i < symbols.Count; i++)
        {
        Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", symbols[i].InstancePath, values[i].
ToString(), values[i].GetType().Name);
        }

        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(client, symbols);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        var resultSumWrite = await writeCommand.WriteAsync(writeValues, cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumSymbolWrite [▶ 1248]

6.5.11.1 SumSymbolRead Constructor

Initializes a new instance of the SumSymbolRead [▶ 1242] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SumSymbolRead(
    IAdsConnection connection,
    IList<ISymbol> symbols
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The ADS Connection.
symbols	Type: System.Collections.Generic.IList.ISymbol [▶ 2176]. The symbols to read

Reference

SumSymbolRead Class [▶ 1242]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.11.2 SumSymbolRead Methods

The SumSymbolRead [▶ 1242] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Read [▶ 1245]	Reads the Values.
	ReadAsync [▶ 1246]	Reads Symbol values as an asynchronous operation.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryRead [▶ 1247]	Reads the specified symbols.

Reference

SumSymbolRead Class [▶ 1242]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.11.2.1 SumSymbolRead.Read Method

Reads the Values.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object[] Read()
```

Return Value

Type: .Object.
System.Object[].

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumSymbolRead failed!

Remarks

The return values are automatically marshalled to their appropriate .NET types.

Reference

SumSymbolRead Class [▶ 1242]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolRead.TryRead(.Object.., .AdsErrorCode..) [▶ 1247]

SumSymbolRead.ReadAsync(CancellationToken) [▶ 1246]

6.5.11.2.2 SumSymbolRead.ReadAsync Method

Reads Symbol values as an asynchronous operation.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumValues> ReadAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultSumValues [▶ 1220].
An asynchronous task that represents the 'Read' operation and returns a ResultSumValues [▶ 1220]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumSymbolRead Class [▶ 1242]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolRead.Read. [▶ 1245]

SumSymbolRead.TryRead(.Object.., .AdsErrorCode..) [▶ 1247]

6.5.11.2.3 SumSymbolRead.TryRead Method

Reads the specified symbols.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryRead(
    out Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values	Type: .System.Object.. The values.
returnCodes	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575].. The return codes.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Remarks

The returned values are automatically marshalled to their appropriate .NET types.

Reference

SumSymbolRead Class [▶ 1242]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolRead.Read. [▶ 1245]

SumSymbolRead.ReadAsync(CancellationToken) [▶ 1246]

6.5.12 SumSymbolWrite Class

Class for ADS Sum symbolic Write Access.

Inheritance Hierarchy

System.Object
  SumCommandWrapper.SumWrite.
    SumSymbolCommand.SumWrite.
      TwinCAT.Ads.SumCommand.SumSymbolWrite
Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SumSymbolWrite : SumSymbolCommand<SumWrite>
```

The SumSymbolWrite type exposes the following members.

Constructors

	Name	Description
	SumSymbolWrite [▶ 1250]	Initializes a new instance of the SumSymbolWrite class.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryWrite [▶ 1251]	Writes the specified values.
	Write [▶ 1252]	Writes the specified values.
	WriteAsync [▶ 1253]	Reads the symbol values asynchronously.

Remarks

The SumSymbolWrite implements symbolic write access with automatic (dynamic) value marshalling. The advantage of the symbolic access is (in contrast to the handle access classes SumHandleRead [▶ 1228],SumHandleWrite [▶ 1232]) that all type information is available when using this ADS Sum Command. The disadvantage is, that the Symbolic information must be loaded beforehand, see examples.

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsSession

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsSession session = new AdsSession(address))
    {
    // Connect to the device target.
    AdsConnection connection = (AdsConnection)session.Connect();

    // Load symbolic information
    var resultSymbols = await session.SymbolServer.GetSymbolsAsync(cancel);
    resultSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> symbols = resultSymbols.Symbols;

    ISymbol bVar1 = symbols["GVL.bVar1"];
    ISymbol bVar2 = symbols["GVL.bVar2"];
    ISymbol projectName = symbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];

    SymbolCollection coll = new SymbolCollection() {bVar1, bVar2, projectName};

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(connection,coll);
    var resultReadValues = await readCommand.ReadAsync(cancel);

    if (resultReadValues.Succeeded)
    {
        object[] values = resultReadValues.Values;

        for (int i = 0; i < coll.Count; i++)
        {
        Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", coll[i].InstancePath, values[i].ToS
tring(), values[i].GetType().Name);
        }
    }
    // Sum Command Write

    SumSymbolWrite writeCommand = new SumSymbolWrite(connection,coll);
    object[] writeValues = new object[] {true, (short) 42, "MyNewProjectName"};

    var resultWrite = await writeCommand.WriteAsync(writeValues,cancel);
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Examples

Usage of SumSymbolRead/SumSymbolWrite with AdsClient

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for start:");
    Console.ReadLine();
```

```
    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    // Parse the command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
    // Connect the AdsClient to the device target.
    client.Connect(address);

    // Load symbolic information
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);
    var resultReadSymbols = await loader.GetSymbolsAsync(cancel);

    resultReadSymbols.ThrowOnError();

    ISymbolCollection<ISymbol> allSymbols = resultReadSymbols.Symbols;

    ISymbol bVar1 = allSymbols["GVL.bVar1"];
    ISymbol bVar2 = allSymbols["GVL.iCount"];
    ISymbol projectName = allSymbols["TwinCAT_SystemInfoVarList._AppInfo.ProjectName"];

    SymbolCollection symbols = new SymbolCollection() {bVar1, bVar2, projectName};

    // Sum Command Read
    SumSymbolRead readCommand = new SumSymbolRead(client,symbols);
    var resultSumRead = await readCommand.ReadAsync(cancel);

    if (resultSumRead.Succeeded)
    {
        object[] values = resultSumRead.Values;

        for (int i = 0; i < symbols.Count; i++)
        {
        Console.WriteLine("Symbol: {0} (Value: {1}, Type: {2})", symbols[i].InstancePath, values[i].
ToString(), values[i].GetType().Name);
        }

        // Sum Command Write
        SumSymbolWrite writeCommand = new SumSymbolWrite(client, symbols);
        object[] writeValues = new object[] { true, (short)42, "MyNewProjectName" };

        var resultSumWrite = await writeCommand.WriteAsync(writeValues, cancel);
    }
    }

    Console.WriteLine("");
    Console.WriteLine("Press [Enter] for leave:");
    Console.ReadLine();
}
```

Reference

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

TwinCAT.Ads.SumCommand.ISumCommand [▶ 1206]

TwinCAT.Ads.SumCommand.SumSymbolWrite

6.5.12.1 SumSymbolWrite Constructor

Initializes a new instance of the SumSymbolWrite [▶ 1248] class.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version:
5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SumSymbolWrite(
    IAdsConnection connection,
    IList<ISymbol> symbols
)
```

Parameters

connection	Type: TwinCAT.Ads.IAdsConnection [▶ 765] The ADS Connection.
symbols	Type: System.Collections.Generic.IList.ISymbol [▶ 2176]. The symbols to read

Reference

SumSymbolWrite Class [▶ 1248]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.12.2 SumSymbolWrite Methods

The SumSymbolWrite [▶ 1248] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryWrite [▶ 1251]	Writes the specified values.
	Write [▶ 1252]	Writes the specified values.
	WriteAsync [▶ 1253]	Reads the symbol values asynchronously.

Reference

SumSymbolWrite Class [▶ 1248]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

6.5.12.2.1 SumSymbolWrite.TryWrite Method

Writes the specified values.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AdsErrorCode TryWrite(
    Object[] values,
    out AdsErrorCode[] returnCodes
)
```

Parameters

values	Type: .System.Object. The values.
returnCodes	Type: .TwinCAT.Ads.AdsErrorCode [▶ 575].. The return codes.

Return Value

Type: AdsErrorCode [▶ 575]
AdsErrorCode.

Remarks

The written values will be marshalled automatically to their appropriate ADS types.

Reference

SumSymbolWrite Class [▶ 1248]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolWrite.Write(.Object.) [▶ 1252]

SumSymbolWrite.WriteAsync(.Object., CancellationToken) [▶ 1253]

6.5.12.2.2 SumSymbolWrite.Write Method

Writes the specified values.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write(
    Object[] values
)
```

Parameters

values	Type: .System.Object. The values.

Exceptions

Exception	Condition
AdsSumCommandException [▶ 632]	SumSymbolWrite failed!

Remarks

The values will be marshalled automatically to their appropriate ADS types.

Reference

SumSymbolWrite Class [▶ 1248]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolWrite.TryWrite(.Object., .AdsErrorCode..) [▶ 1251]

SumSymbolWrite.WriteAsync(.Object., CancellationToken) [▶ 1253]

6.5.12.2.3 SumSymbolWrite.WriteAsync Method

Reads the symbol values asynchronously.

Namespace: TwinCAT.Ads.SumCommand [▶ 1205]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultSumCommand> WriteAsync(
    Object[] values,
    CancellationToken cancel
)
```

Parameters

values	Type: .System.Object. The values.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultSumCommand [▶ 1210].
An asynchronous task that represents the 'Write' operation and returns a ResultSumCommand [▶ 1210]. The overall error return code is contained in the ErrorCode [▶ 992] property.

Reference

SumSymbolWrite Class [▶ 1248]

TwinCAT.Ads.SumCommand Namespace [▶ 1205]

SumSymbolWrite.Write(.Object.) [▶ 1252]

SumSymbolWrite.TryWrite(.Object., .AdsErrorCode..) [▶ 1251]

6.6 TwinCAT.Ads.TcpRouter Namespace

Classes

	Class	Description
	AmsTcpIpRouter [▶ 1254]	ADS Router class
	Route [▶ 1281]	Class Route specifies a an AMS/ADS Route object.
	RouteCollection [▶ 1292]	Collection of routes.
	RouterException [▶ 1306]	An RouterException [▶ 1306] is thrown on communication errors in the AmsTcpIpRouter [▶ 1254] class.
	RouterNotInitializedException [▶ 1310]	Class RouterNotInitializedException. Implements the RouterException [▶ 1306]
	RouterNotStartedException [▶ 1313]	Class RouterNotStartedException. Implements the RouterException [▶ 1306]
	RouterStatusChangedEventArgs [▶ 1317]	Event Arguments sent when the RouterStatusChanged [▶ 1281]. Implements the EventArgs
	StaticRoutesXmlConfigurationBuilderExtension [▶ 1320]	Extension class adding StaticRoutes.xml file reading to the IConfigurationBuilder.
	StaticRoutesXmlConfigurationProvider [▶ 1321]	Class StaticRoutesXmlConfigurationProvider. Implements the ConfigurationProvider
	StaticRoutesXmlConfigurationSource [▶ 1326]	StaticRoutes Configuration Sources Implements the IConfigurationSource

Interfaces

	Interface	Description
	IAmsRouter [▶ 1273]	Interface IAmsRouter

Enumerations

	Enumeration	Description
	RouterStatus [▶ 1316]	Enum RouterStatus

6.6.1 AmsTcpIpRouter Class

ADS Router class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.AmsTcpIpRouter
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AmsTcpIpRouter : IAmsRouter
```

The AmsTcpIpRouter type exposes the following members.

Constructors

	Name	Description
	AmsTcpIpRouter(IConfiguration) [▶ 1260]	Initializes a new instance of the AmsTcpIpRouter class.
	AmsTcpIpRouter(AmsNetId) [▶ 1261]	Initializes a new instance of the AmsTcpIpRouter class.
	AmsTcpIpRouter(ILogger, IConfiguration) [▶ 1261]	Initializes a new instance of the AmsTcpIpRouter class.
	AmsTcpIpRouter(AmsNetId, Int32, IPAddress, Int32, ILogger) [▶ 1262]	Initializes a new instance of the AmsTcpIpRouter class.

Properties

	Name	Description
	IsActive [▶ 1263]	Gets a value indicating whether the IAmsRouter [▶ 1273] is active (Running or in Starting / Stopping state).
	IsRunning [▶ 1264]	Gets a value indicating whether the IAmsRouter [▶ 1273] is running (Start phase completely finished).
S	LocalNetId [▶ 1264]	Gets the local AmsNetId [▶ 665]
	Loopback [▶ 1265]	Gets the loopback alias
	NetId [▶ 1265]	The Local AmsNetId [▶ 665] of this router.
	RouterStatus [▶ 1266]	Gets the router status.

Methods

	Name	Description
	AddRoute [▶ 1267]	Adds a dynamic Route [▶ 1281]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IsRegisteredServer [▶ 1267]	Determines whether the Address specifies an locally registered AmsServer
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	RemoveRoute(String) [▶ 1268]	Removes a dynamic Route [▶ 1281]
	RemoveRoute(AmsNetId) [▶ 1269]	Removes a dynamic Route [▶ 1281]
	StartAsync [▶ 1270]	Starts the AmsTcpIpRouter asynchronously.
	Stop [▶ 1270]	Stops the AmsTcpIpRouter.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddRoute [▶ 1271]	Adds a dynamic Route [▶ 1281]

Events

	Name	Description
	RouterStatusChanged [▶ 1272]	Occurs when the router status changes.

Fields

	Name	Description
	DEFAULT_TCP_PORT [▶ 1273]	The default TCP port (0xBF02, 48898)
	TCP_TIMEOUT [▶ 1273]	The TCP timeout

Remarks

The AmsTcpIpRouter implements a simple ADS/AMS Router for systems, where no TwinCAT Router installation is available. It opens by default an TCP Loopback port (127.0.0.1) Port BF02 (DEFAULT_TCP_PORT [▶ 1273]) and routes the ADS/AMS communication between AdsServer and AdsClient. If the addressed target server is not on the local system, the AMS Frames are addressed to remote systems, if the route is appropriately assigned on source and target system (on both sides usually via StaticRoutes.xml).

Property Name	Description	Default
Name	The name of the local system.	
NetId	The AmsNetId of the local system.	
RemoteConnections	List of the registered remote Routes.	None/Empty
TcpPort	The TCP Port to be used for external ADS communication.	0xBF02 (48898)
LoopbackIP	The LoopbackIP of Device internal communication. This setting should only be changed if ADS Server and Router application should run on different (virtual) devices. On the AdsServer side an appropriate system configuration via RouterEndPoint [▶ 1546] is necessary if the loopback ip is changed.	Loopback 127.0.0.1
LoopbackPort	The Loopback Port of Device internal ADS communication. This setting should only be changed when different TcpPorts are necessary to seperate internal and external router communcation (e.g. in WSL2 docker containers) where the LoopbackIP 127.0.0.1 isn't appropiate. On the AdsServer side an appropriate system configuration via RouterEndPoint [▶ 1546] is necessary if the loopback port is changed.	0xBF02 (48898)

The router settings can be configured by the .NET Configuration Builder IConfigurationBuilder or simply by calling one of the AmsTcpIpRouter constructor overloads.

Xml Configuration (StaticRoutes.xml)	AddStaticRoutesXmlConfiguration(IConfigurationBuilder) [▶ 1320]
Json Configuration (appSettings.json)	[!:Microsoft.Extensions.Configuration.JsonConfigurationExtensions.AddJsonFile(string)]
Environment variables	[!:Microsoft.Extensions.Configuration.EnvironmentVariablesExtensions.AddEnvironmentVariables()]
ConsoleArguments	[!:Microsoft.Extensions.Configuration.CommandLineConfigurationExtensions.AddCommandLine(string[])]

Examples

The following sample shows how to use the AmsTcpIpRouter class within an own Console application. This console application can also be accessed as binary from the Nuget.org package repository. The ID of the package is 'Beckhoff.TwinCAT.Ads.AdsRouterConsole'.

Ads Router WorkerService

```
class Program
{
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
public static void Main(string[] args)
{
    try
    {
    CreateHostBuilder(args).Build().Run();
```

```
        }
        catch (OperationCanceledException /*cex*/)
        {
        Console.WriteLine("Router cancelled!");
        }
        catch (Exception ex)
        {
        Console.WriteLine($"Router failed with '{ex.Message}'");
        }
    }

    /// <summary>
    /// Creates the host builder.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>IHostBuilder.</returns>
    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
        .ConfigureServices((hostContext, services) =>
        {
            services.AddHostedService<RouterService>();
        })
        .ConfigureAppConfiguration((hostingContext, config) =>
        {
            // Add further AppConfigurationProvider here.
            //config.Sources.Clear(); // Clear all default config sources
            //config.AddEnvironmentVariables(""); // Use Environment variables
            //config.AddCommandLine(args); // Use Command Line
            //config.AddJsonFile("appSettings.json"); // Use Appsettings
            //
config.AddStaticRoutesXmlConfiguration(); // Overriding settings with StaticRoutes.Xml
        })
        .ConfigureLogging(logging =>
        {
            //logging.ClearProviders();
            // Adding console logging here.
            //logging.AddConsole();
        })
        ;
    }

    /// <summary>
    /// The RouterService instance represents a long running (hosted) service that implements an <se
e cref="AmsTcpIpRouter"/>.
    /// Implements the <see cref="Microsoft.Extensions.Hosting.BackgroundService" />
    /// </summary>
    /// <remarks>
    /// Long running Background task that runs a <see cref="AmsTcpIpRouter."/>.
    /// The service is stopped via the <see cref="CancellationToken"/
> given to the <see cref="RouterService.ExecuteAsync(CancellationToken)"/> method.
    /// </remarks>
    /// <seealso cref="Microsoft.Extensions.Hosting.BackgroundService" />
    public class RouterService : BackgroundService
    {
    /// <summary>
    /// Logger
    /// </summary>
    private readonly ILogger<RouterService> _logger;
    /// <summary>
    /// Configuration
    /// </summary>
    private readonly IConfiguration _configuration;

    /// <summary>
    /// Initializes a new instance of the <see cref="RouterService"/> class.
    /// </summary>
    /// <param name="logger">The logger.</param>
    /// <param name="configuration">The configuration.</param>
    public RouterService(ILogger<RouterService> logger, IConfiguration configuration)
    {
        _logger = logger;
        _configuration = configuration;
        string value = (string)_configuration.GetValue("ASPNETCORE_ENVIRONMENT", "Production");
    }

    /// <summary>
    /// Execute the Router asynchronously as <see cref="BackgroundService"/>.
    /// </summary>
    /// <param name="cancel">The cancellation token.</param>
    protected override async Task ExecuteAsync(CancellationToken cancel)
```

```
    {
        AmsTcpIpRouter router;

        using (_logger.BeginScope("Starting"))
        {
        StringBuilder appCommon = new StringBuilder();

        appCommon.AppendLine($"ApplicationPath: {Environment.GetCommandLineArgs()[0]}");
        appCommon.AppendLine($"BaseDirectory: {System.AppContext.BaseDirectory}");
        appCommon.AppendLine($"CurrentDirectory: {Directory.GetCurrentDirectory()}");
        //_logger.LogInformation(sB.ToString());

        StringBuilder config= new StringBuilder();
        string value = (string)_configuration.GetValue("ASPNETCORE_ENVIRONMENT", "Production");
        config.AppendLine($"ASPNETCORE_ENVIRONMENT: {value}");

        Console.WriteLine("Application Directories");
        Console.WriteLine("======================");
        Console.WriteLine(appCommon);
        Console.WriteLine("");
        Console.WriteLine("Configuration");
        Console.WriteLine("=============");
        Console.WriteLine(config);
        Console.WriteLine("");

        Console.WriteLine("Press Ctrl + C to shutdown!");

        router = new AmsTcpIpRouter(_logger, _configuration);
        router.RouterStatusChanged += Router_RouterStatusChanged;

        // Use this overload to instantiate a Router without support of IHost/
IConfigurationProvider support and parametrize by code
        // AmsTcpIpRouter router = new AmsTcpIpRouter(new AmsNetId("1.2.3.4.5.6"), AmsTcpIpRouter.DE
FAULT_TCP_PORT,IPAddress.Loopback,AmsTcpIpRouter.DEFAULT_TCP_PORT,_logger);
        // router.AddRoute(...);

        _logger.LogInformation(appCommon.ToString());
        _logger.LogInformation(config.ToString());
        }

        Task routerTask = router.StartAsync(cancel); // Start the router

#if ADSSERVER
        // AdsServer could be started here:
        TestAdsServer testServer = new TestAdsServer(_logger);
        _logger.LogInformation($"Adding Test Server on port '{testServer.ServerAddress}'");
        Task systemServiceTask = testServer.ConnectServerAndWaitAsync(cancel);

        await Task.WhenAll(routerTask, systemServiceTask);
#else
        await routerTask;
#endif
        }

    /// <summary>
    /// Handles the RouterStatusChanged event of the <see cref="AmsTcpIpRouter"/>
    /// </summary>
    /// <param name="sender">The source of the event.</param>
    /// <param name="e">The <see cref="RouterStatusChangedEventArgs"/
> instance containing the event data.</param>
    private void Router_RouterStatusChanged(object? sender, RouterStatusChangedEventArgs e)
    {
        if (e.RouterStatus == RouterStatus.Started)
        {
        // From here on, the Router is available to receive Data.
        }
    }
    }
```

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

RouteConfig

TwinCAT.Ams.AmsConfiguration [▶ 1543]

6.6.1.1 AmsTcpIpRouter Constructor

Overload List

	Name	Description
	AmsTcpIpRouter(IConfiguration) [▶ 1260]	Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.
	AmsTcpIpRouter(AmsNetId) [▶ 1261]	Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.
	AmsTcpIpRouter(ILogger, IConfiguration) [▶ 1261]	Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.
	AmsTcpIpRouter(AmsNetId, Int32, IPAddress, Int32, ILogger) [▶ 1262]	Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.1.1 AmsTcpIpRouter Constructor (IConfiguration)

Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsTcpIpRouter(
    IConfiguration configuration
)
```

Parameters

configuration	Type: IConfiguration

Remarks

The AmsTcpIpRouter [▶ 1254] will be connected to the Default port DEFAULT_TCP_PORT [▶ 1273].

Reference

AmsTcpIpRouter Class [▶ 1254]

AmsTcpIpRouter Overload [▶ 1260]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.1.2 AmsTcpIpRouter Constructor (AmsNetId)

Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsTcpIpRouter(
    AmsNetId local
)
```

Parameters

local	Type: TwinCAT.Ads.AmsNetId [▶ 665] The local.

Remarks

The AmsTcpIpRouter [▶ 1254] will be connected to the Default port DEFAULT_TCP_PORT [▶ 1273].

Reference

AmsTcpIpRouter Class [▶ 1254]

AmsTcpIpRouter Overload [▶ 1260]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.1.3 AmsTcpIpRouter Constructor (ILogger, IConfiguration)

Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsTcpIpRouter(
    ILogger logger,
    IConfiguration configuration
)
```

Parameters

logger	Type: ILogger The logger.
configuration	Type: IConfiguration The configuration.

Exceptions

Exception	Condition
Exception	No IPv4 address for server

Remarks

The AmsTcpIpRouter [▶ 1254] will be connected to the Default port DEFAULT_TCP_PORT [▶ 1273].

Reference

AmsTcpIpRouter Class [▶ 1254]

AmsTcpIpRouter Overload [▶ 1260]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.1.4 AmsTcpIpRouter Constructor (AmsNetId, Int32, IPAddress, Int32, ILogger)

Initializes a new instance of the AmsTcpIpRouter [▶ 1254] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsTcpIpRouter(
    AmsNetId local,
    int externalPort,
    IPAddress loopbackIP,
    int loopbackPort,
    ILogger logger
)
```

Parameters

local	Type: TwinCAT.Ads.AmsNetId [▶ 665] The (enforced) local NET Id.
externalPort	Type: System.Int32 The TCP/IP port the AmsTcpIpRouter [▶ 1254] is using.
loopbackIP	Type: System.Net.IPAddress The loopback ip.
loopbackPort	Type: System.Int32 The loopback port.
logger	Type: ILogger The logger.

Remarks

This constructor doesn't read the 'StaticRoutes.xml'. Instead, it is expected to force the local AmsNetId [▶ 665], the TCP/IP port and add the routes as Dynamic routes.

Reference

AmsTcpIpRouter Class [▶ 1254]

AmsTcpIpRouter Overload [▶ 1260]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2 AmsTcpIpRouter Properties

The AmsTcpIpRouter [▶ 1254] type exposes the following members.

Properties

	Name	Description
	IsActive [▶ 1263]	Gets a value indicating whether the IAmsRouter [▶ 1273] is active (Running or in Starting / Stopping state).
	IsRunning [▶ 1264]	Gets a value indicating whether the IAmsRouter [▶ 1273] is running (Start phase completely finished).
	LocalNetId [▶ 1264]	Gets the local AmsNetId [▶ 665]
	Loopback [▶ 1265]	Gets the loopback alias
	NetId [▶ 1265]	The Local AmsNetId [▶ 665] of this router.
	RouterStatus [▶ 1266]	Gets the router status.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.1 AmsTcpIpRouter.IsActive Property

Gets a value indicating whether the IAmsRouter [▶ 1273] is active (Running or in Starting / Stopping state).

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsActive { get; }
```

Property Value

Type: Boolean
true if this the router is active; otherwise, false.

Implements

IAmsRouter.IsActive [▶ 1275]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.2 AmsTcpIpRouter.IsRunning Property

Gets a value indicating whether the IAmsRouter [▶ 1273] is running (Start phase completely finished).

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsRunning { get; }
```

Property Value

Type: Boolean
true if this the router is running; otherwise, false.

Implements

IAmsRouter.IsRunning [▶ 1275]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.3 AmsTcpIpRouter.LocalNetId Property

Gets the local AmsNetId [▶ 665]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static AmsNetId LocalNetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The local net identifier.

Remarks

This is initialized during the execution of the AmsTcpIpRouter [▶ 1254] constructor..

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.4 AmsTcpIpRouter.Loopback Property

Gets the loopback alias

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IPAddress Loopback { get; }
```

Property Value

Type: IPAddress
The loopback.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.5 AmsTcpIpRouter.NetId Property

The Local AmsNetId [▶ 665] of this router.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The net identifier.

Implements

IAmsRouter.NetId [▶ 1276]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.2.6 AmsTcpIpRouter.RouterStatus Property

Gets the router status.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RouterStatus RouterStatus { get; }
```

Property Value

Type: RouterStatus [▶ 1316]
The router status.

Implements

IAmsRouter.RouterStatus [▶ 1276]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

AmsTcpIpRouter.RouterStatusChanged [▶ 1272]

6.6.1.3 AmsTcpIpRouter Methods

The AmsTcpIpRouter [▶ 1254] type exposes the following members.

Methods

	Name	Description
	AddRoute [▶ 1267]	Adds a dynamic Route [▶ 1281]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IsRegisteredServer [▶ 1267]	Determines whether the Address specifies an locally registered AmsServer
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	RemoveRoute(String) [▶ 1268]	Removes a dynamic Route [▶ 1281]

	Name	Description
	RemoveRoute(AmsNetId) [▶ 1269]	Removes a dynamic Route [▶ 1281]
	StartAsync [▶ 1270]	Starts the AmsTcpIpRouter [▶ 1254] asynchronously.
	Stop [▶ 1270]	Stops the AmsTcpIpRouter [▶ 1254].
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAddRoute [▶ 1271]	Adds a dynamic Route [▶ 1281]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.1 AmsTcpIpRouter.AddRoute Method

Adds a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void AddRoute(
    Route route
)
```

Parameters

route	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The route.

Implements

IAmsRouter.AddRoute(Route) [▶ 1277]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.2 AmsTcpIpRouter.IsRegisteredServer Method

Determines whether the Address specifies an locally registered AmsServer

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsRegisteredServer(
    AmsAddress serverAddress
)
```

Parameters

serverAddress	Type: TwinCAT.Ads.AmsAddress [▶ 648] The server address.

Return Value

Type: Boolean
true if [is server registered] [the specified server address]; otherwise, false.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.3 AmsTcpIpRouter.RemoveRoute Method

Overload List

	Name	Description
	RemoveRoute(String) [▶ 1268]	Removes a dynamic Route [▶ 1281]
	RemoveRoute(AmsNetId) [▶ 1269]	Removes a dynamic Route [▶ 1281]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

AmsTcpIpRouter.RemoveRoute Method (String)

Removes a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool RemoveRoute(
    string route
)
```

Parameters

route	Type: System.String The route.

Return Value

Type: Boolean
true if removed, false otherwise.

Implements

IAmsRouter.RemoveRoute(String) [▶ 1278]

Reference

AmsTcpIpRouter Class [▶ 1254]

RemoveRoute Overload [▶ 1268]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

AmsTcpIpRouter.RemoveRoute Method (AmsNetId)

Removes a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool RemoveRoute(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The NetId of the route.

Return Value

Type: Boolean
true if removed, false otherwise.

Implements

IAmsRouter.RemoveRoute(AmsNetId) [▶ 1278]

Reference

AmsTcpIpRouter Class [▶ 1254]

RemoveRoute Overload [▶ 1268]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.4 AmsTcpIpRouter.StartAsync Method

Starts the AmsTcpIpRouter [▶ 1254] asynchronously.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task StartAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task
Task.

Implements

IAmsRouter.StartAsync(CancellationToken) [▶ 1279]

Remarks

The asynchronous task runs as long until Stop. [▶ 1270]is called.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.5 AmsTcpIpRouter.Stop Method

Stops the AmsTcpIpRouter [▶ 1254].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Stop()
```

Implements

IAmsRouter.Stop. [▶ 1280]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.3.6 AmsTcpIpRouter.TryAddRoute Method

Adds a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryAddRoute(
    Route route
)
```

Parameters

route	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The route.

Return Value

Type: Boolean
true if added, false otherwise.

Implements

IAmsRouter.TryAddRoute(Route) [▶ 1280]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.4 AmsTcpIpRouter Events

The AmsTcpIpRouter [▶ 1254] type exposes the following members.

Events

	Name	Description
⚡	RouterStatusChanged [▶ 1272]	Occurs when the router status changes.

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.4.1 AmsTcpIpRouter.RouterStatusChanged Event

Occurs when the router status changes.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<RouterStatusChangedEventArgs> RouterStatusChanged
```

Value

Type: System.EventHandler.RouterStatusChangedEventArgs [▶ 1317].

Implements

IAmsRouter.RouterStatusChanged [▶ 1281]

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

AmsTcpIpRouter.RouterStatus [▶ 1266]

6.6.1.5 AmsTcpIpRouter Fields

The AmsTcpIpRouter [▶ 1254] type exposes the following members.

Fields

	Name	Description
	DEFAULT_TCP_PORT [▶ 1273]	The default TCP port (0xBF02, 48898)
	TCP_TIMEOUT [▶ 1273]	The TCP timeout

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.5.1 AmsTcpIpRouter.DEFAULT_TCP_PORT Field

The default TCP port (0xBF02, 48898)

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public const int DEFAULT_TCP_PORT = 48898
```

Field Value

Type: Int32

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.1.5.2 AmsTcpIpRouter.TCP_TIMEOUT Field

The TCP timeout

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static int TCP_TIMEOUT
```

Field Value

Type: Int32

Reference

AmsTcpIpRouter Class [▶ 1254]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2 IAmsRouter Interface

Interface IAmsRouter

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAmsRouter
```

The IAmsRouter type exposes the following members.

Properties

	Name	Description
	IsActive [▶ 1275]	Gets a value indicating whether the IAmsRouter is active (Running or in Starting / Stopping state).
	IsRunning [▶ 1275]	Gets a value indicating whether the IAmsRouter is running (Start phase completely finished).
	NetId [▶ 1276]	The Local AmsNetId [▶ 665] of the IAmsRouter
	RouterStatus [▶ 1276]	Gets the router status.

Methods

	Name	Description
	AddRoute [▶ 1277]	Adds a dynamic Route [▶ 1281]
	RemoveRoute(String) [▶ 1278]	Removes a dynamic Route [▶ 1281]
	RemoveRoute(AmsNetId) [▶ 1278]	Removes a dynamic Route [▶ 1281]
	StartAsync [▶ 1279]	Starts the IAmsRouter asynchronously.
	Stop [▶ 1280]	Stops the IAmsRouter.
	TryAddRoute [▶ 1280]	Adds a dynamic Route [▶ 1281]

Events

	Name	Description
	RouterStatusChanged [▶ 1281]	Occurs when the router status changes.

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.1 IAmsRouter Properties

The IAmsRouter [▶ 1273] type exposes the following members.

Properties

	Name	Description
	IsActive [▶ 1275]	Gets a value indicating whether the IAmsRouter [▶ 1273] is active (Running or in Starting / Stopping state).
	IsRunning [▶ 1275]	Gets a value indicating whether the IAmsRouter [▶ 1273] is running (Start phase completely finished).
	NetId [▶ 1276]	The Local AmsNetId [▶ 665] of the IAmsRouter [▶ 1273]
	RouterStatus [▶ 1276]	Gets the router status.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.1.1 IAmsRouter.IsActive Property

Gets a value indicating whether the IAmsRouter [▶ 1273] is active (Running or in Starting / Stopping state).

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsActive { get; }
```

Property Value

Type: Boolean
true if this the router is active; otherwise, false.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.1.2 IAmsRouter.IsRunning Property

Gets a value indicating whether the IAmsRouter [▶ 1273] is running (Start phase completely finished).

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsRunning { get; }
```

Property Value

Type: Boolean
true if this the router is running; otherwise, false.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.1.3 IAmsRouter.NetId Property

The Local AmsNetId [▶ 665] of the IAmsRouter [▶ 1273]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The net identifier.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.1.4 IAmsRouter.RouterStatus Property

Gets the router status.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
RouterStatus RouterStatus { get; }
```

Property Value

Type: RouterStatus [▶ 1316]
The router status.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2 IAmsRouter Methods

The IAmsRouter [▶ 1273] type exposes the following members.

Methods

	Name	Description
	AddRoute [▶ 1277]	Adds a dynamic Route [▶ 1281]
	RemoveRoute(String) [▶ 1278]	Removes a dynamic Route [▶ 1281]
	RemoveRoute(AmsNetId) [▶ 1278]	Removes a dynamic Route [▶ 1281]
	StartAsync [▶ 1279]	Starts the IAmsRouter [▶ 1273] asynchronously.
	Stop [▶ 1280]	Stops the IAmsRouter [▶ 1273].
	TryAddRoute [▶ 1280]	Adds a dynamic Route [▶ 1281]

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2.1 IAmsRouter.AddRoute Method

Adds a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void AddRoute(
    Route route
)
```

Parameters

route	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The route.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2.2 IAmsRouter.RemoveRoute Method

Overload List

	Name	Description
	RemoveRoute(String) [▶ 1278]	Removes a dynamic Route [▶ 1281]
	RemoveRoute(AmsNetId) [▶ 1278]	Removes a dynamic Route [▶ 1281]

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

IAmsRouter.RemoveRoute Method (String)

Removes a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool RemoveRoute(
    string route
)
```

Parameters

route	Type: System.String The route.

Return Value

Type: Boolean
true if removed, false otherwise.

Reference

IAmsRouter Interface [▶ 1273]

RemoveRoute Overload [▶ 1278]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

IAmsRouter.RemoveRoute Method (AmsNetId)

Removes a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool RemoveRoute(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The NetId of the route.

Return Value

Type: Boolean
true if removed, false otherwise.

Reference

IAmsRouter Interface [▶ 1273]

RemoveRoute Overload [▶ 1278]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2.3 IAmsRouter.StartAsync Method

Starts the IAmsRouter [▶ 1273] asynchronously.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task StartAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task
Task.

Remarks

The asynchronous task runs as long until Stop. [▶ 1280]is called.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2.4 IAmsRouter.Stop Method

Stops the IAmsRouter [▶ 1273].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Stop()
```

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.2.5 IAmsRouter.TryAddRoute Method

Adds a dynamic Route [▶ 1281]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryAddRoute(
    Route route
)
```

Parameters

route	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The route.

Return Value

Type: Boolean
true if added, false otherwise.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.3 IAmsRouter Events

The IAmsRouter [▶ 1273] type exposes the following members.

Events

	Name	Description
⚡	RouterStatusChanged [▶ 1281]	Occurs when the router status changes.

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.2.3.1 IAmsRouter.RouterStatusChanged Event

Occurs when the router status changes.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<RouterStatusChangedEventArgs> RouterStatusChanged
```

Value

Type: System.EventHandler.RouterStatusChangedEventArgs [▶ 1317].

Reference

IAmsRouter Interface [▶ 1273]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3 Route Class

Class Route specifies a an AMS/ADS Route object.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.Route
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Route
```

The Route type exposes the following members.

Constructors

	Name	Description
	Route(String, AmsNetId, IList.IPAddress.) [▶ 1283]	Initializes a new instance of the Route class.
	Route(String, AmsNetId, String) [▶ 1284]	Initializes a new instance of the Route class.

Properties

	Name	Description
	Address [▶ 1285]	Gets the address of the Route.
	HostName [▶ 1285]	Gets the Host name of the route.
	IPAddresses [▶ 1286]	Gets the resolved IP Addresses belonging to this Route.
	IsHostNameAddressed [▶ 1286]	Gets a value indicating whether this instance is an HostName Route.
	IsIPAddressed [▶ 1287]	Gets a value indicating whether the Route address specifies an IPAddress
	IsResolved [▶ 1287]	Gets a value indicating whether the IPAddresses are resolved.
	Name [▶ 1288]	Gets the name of the Route.
	NetId [▶ 1288]	Gets the AmsNetId [▶ 665] of the route.

Methods

	Name	Description
	Equals [▶ 1289]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1289]	Returns a hash code for this instance. (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1290]	Returns a String that represents this instance. (Overrides Object.ToString..)

Operators

	Name	Description
	Equality [▶ 1291]	Implements the == operator.
	Inequality [▶ 1291]	Implements the != operator.

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.1 Route Constructor

Overload List

	Name	Description
	Route(String, AmsNetId, IList.IPAddress.) [▶ 1283]	Initializes a new instance of the Route [▶ 1281] class.
	Route(String, AmsNetId, String) [▶ 1284]	Initializes a new instance of the Route [▶ 1281] class.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.1.1 Route Constructor (String, AmsNetId, IList.IPAddress.)

Initializes a new instance of the Route [▶ 1281] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Route(
    string name,
    AmsNetId netId,
    IList<IPAddress> ips
)
```

Parameters

name	Type: System.String The name of the route.

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The net identifier.
ips	Type: System.Collections.Generic.IList.IPAddress. The ips.

Reference

Route Class [▶ 1281]

Route Overload [▶ 1283]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.1.2 Route Constructor (String, AmsNetId, String)

Initializes a new instance of the Route [▶ 1281] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Route(
    string name,
    AmsNetId netId,
    string address
)
```

Parameters

name	Type: System.String The name.
netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The AmsNetId of the route.
address	Type: System.String The address (HostName or IP)

Reference

Route Class [▶ 1281]

Route Overload [▶ 1283]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2 Route Properties

The Route [▶ 1281] type exposes the following members.

Properties

	Name	Description
	Address [▶ 1285]	Gets the address of the Route [▶ 1281].

	Name	Description
	HostName [▶ 1285]	Gets the Host name of the route.
	IPAddresses [▶ 1286]	Gets the resolved IP Addresses belonging to this Route [▶ 1281].
	IsHostNameAddressed [▶ 1286]	Gets a value indicating whether this instance is an HostName Route [▶ 1281].
	IsIPAddressed [▶ 1287]	Gets a value indicating whether the Route address specifies an IPAddress
	IsResolved [▶ 1287]	Gets a value indicating whether the IPAddresses are resolved.
	Name [▶ 1288]	Gets the name of the Route [▶ 1281].
	NetId [▶ 1288]	Gets the AmsNetId [▶ 665] of the route.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.1 Route.Address Property

Gets the address of the Route [▶ 1281].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Address { get; }
```

Property Value

Type: String
The address.

Remarks

The Address can be the IPAddress / or the host name of the target system.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.2 Route.HostName Property

Gets the Host name of the route.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string HostName { get; }
```

Property Value

Type: String
The name of the host.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.3 Route.IPAddresses Property

Gets the resolved IP Addresses belonging to this Route [▶ 1281].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IPAddress[] IPAddresses { get; }
```

Property Value

Type: .IPAddress.
The ip addresses.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.4 Route.IsHostNameAddressed Property

Gets a value indicating whether this instance is an HostName Route [▶ 1281].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsHostNameAddressed { get; }
```

Property Value

Type: Boolean
true if this instance is an HostName Addressed route; otherwise, false.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.5 Route.IsIPAddressed Property

Gets a value indicating whether the Route address specifies an IPAddress

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsIPAddressed { get; }
```

Property Value

Type: Boolean
true If the Address is an IPAddress; otherwise when its specified as HostAddress false.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.6 Route.IsResolved Property

Gets a value indicating whether the IPAddresses are resolved.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsResolved { get; }
```

Property Value

Type: Boolean
true if the IP Addresses are resolved; otherwise, false.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.7 Route.Name Property

Gets the name of the Route [▶ 1281].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.2.8 Route.NetId Property

Gets the AmsNetId [▶ 665] of the route.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsNetId NetId { get; }
```

Property Value

Type: AmsNetId [▶ 665]
The AmsNetId.

Remarks

The AmsNetId [▶ 665] is the unique identifier used for the route.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.3 Route Methods

The Route [▶ 1281] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1289]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1289]	Returns a hash code for this instance. (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1290]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.3.1 Route.Equals Method

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.3.2 Route.GetHashCode Method

Returns a hash code for this instance.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.3.3 Route.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.4 Route Operators

The Route [▶ 1281] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 1291]	Implements the == operator.

	Name	Description
 	Inequality [▶ 1291]	Implements the != operator.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.4.1 Route.Equality Operator

Implements the == operator.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    Route o1,
    Route o2
)
```

Parameters

o1	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The o1.
o2	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.3.4.2 Route.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    Route o1,
    Route o2
)
```

Parameters

o1	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The o1.
o2	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

Route Class [▶ 1281]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4 RouteCollection Class

Collection of routes.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.RouteCollection
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RouteCollection : IList<Route>,
    ICollection<Route>, IEnumerable<Route>, IEnumerable
```

The RouteCollection type exposes the following members.

Constructors

	Name	Description
	RouteCollection. [▶ 1294]	Initializes a new instance of the RouteCollection class.
	RouteCollection(IEnumerable.Route.) [▶ 1294]	Initializes a new instance of the RouteCollection class (copy constructor)

Properties

	Name	Description
	Count [▶ 1295]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1296]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 1296]	Gets or sets the Route [▶ 1281] at the specified index.

Methods

	Name	Description
	Add [▶ 1298]	Adds an item to the ICollection.T..
	AddRange [▶ 1299]	Adds a range of routes.
	Clear [▶ 1299]	Removes all items from the ICollection.T..
	Clone [▶ 1299]	Clones this instance.
	Contains [▶ 1300]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1300]	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1301]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 1302]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1302]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove(String) [▶ 1303]	Removes the specified route from the RouteCollection.
	Remove(AmsNetId) [▶ 1304]	Removes the specified AmsNetId [▶ 665] from the RouteCollection.
	Remove(Route) [▶ 1304]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1305]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAdd [▶ 1306]	Tries to add the route to the RouteCollection.

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.1 RouteCollection Constructor

Overload List

	Name	Description
	RouteCollection. [▶ 1294]	Initializes a new instance of the RouteCollection [▶ 1292] class.
	RouteCollection(IEnumerable.Route.) [▶ 1294]	Initializes a new instance of the RouteCollection [▶ 1292] class (copy constructor)

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.1.1 RouteCollection Constructor

Initializes a new instance of the RouteCollection [▶ 1292] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RouteCollection()
```

Reference

RouteCollection Class [▶ 1292]

RouteCollection Overload [▶ 1294]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.1.2 RouteCollection Constructor (IEnumerable.Route.)

Initializes a new instance of the RouteCollection [▶ 1292] class (copy constructor)

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RouteCollection(
    IEnumerable<Route> routes
)
```

Parameters

routes	Type: System.Collections.Generic.IEnumerable.Route [▶ 1281]. The routes.

Reference

RouteCollection Class [▶ 1292]

RouteCollection Overload [▶ 1294]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.2 RouteCollection Properties

The RouteCollection [▶ 1292] type exposes the following members.

Properties

	Name	Description
	Count [▶ 1295]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1296]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 1296]	Gets or sets the Route [▶ 1281] at the specified index.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.2.1 RouteCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.2.2 RouteCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.2.3 RouteCollection.Item Property

Gets or sets the Route [▶ 1281] at the specified index.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Route this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: Route [▶ 1281]
Route.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3 RouteCollection Methods

The RouteCollection [▶ 1292] type exposes the following members.

Methods

	Name	Description
	Add [▶ 1298]	Adds an item to the ICollection.T..
	AddRange [▶ 1299]	Adds a range of routes.
	Clear [▶ 1299]	Removes all items from the ICollection.T..
	Clone [▶ 1299]	Clones this instance.
	Contains [▶ 1300]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1300]	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1301]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 1302]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1302]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	Remove(String) [▶ 1303]	Removes the specified route from the RouteCollection [▶ 1292].
	Remove(AmsNetId) [▶ 1304]	Removes the specified AmsNetId [▶ 665] from the RouteCollection [▶ 1292].
	Remove(Route) [▶ 1304]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1305]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryAdd [▶ 1306]	Tries to add the route to the RouteCollection [▶ 1292].

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.1 RouteCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    Route item
)
```

Parameters

item	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Exceptions

Exception	Condition
ArgumentException	Cannot add item

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.2 RouteCollection.AddRange Method

Adds a range of routes.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void AddRange(
    IEnumerable<Route> routes
)
```

Parameters

routes	Type: System.Collections.Generic.IEnumerable.Route [▶ 1281]. The routes.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.3 RouteCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.4 RouteCollection.Clone Method

Clones this instance.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RouteCollection Clone()
```

Return Value

Type: RouteCollection [▶ 1292]
RouteCollection.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.5 RouteCollection.Contains Method

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    Route item
)
```

Parameters

item	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The object to locate in the ICollection.T..

Return Value

Type: Boolean
. if item is found in the ICollection.T.; otherwise, ..

Implements

ICollection.T..Contains(T)

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.6 RouteCollection.CopyTo Method

Copies the elements of the ICollection.T. to an Array, starting at a particular Array index.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    Route[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.Ads.TcpRouter.Route [▶ 1281]. The one-dimensional Array that is the destination of the elements copied from ICollection.T.. The Array must have zero-based indexing.
arrayIndex	Type: System.Int32 The zero-based index in array at which copying begins.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.7 RouteCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<Route> GetEnumerator()
```

Return Value

Type: IEnumerator.Route [▶ 1281].
An enumerator that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.8 RouteCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    Route item
)
```

Parameters

item	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.9 RouteCollection.Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    Route item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Exceptions

Exception	Condition
ArgumentException	Cannot add item

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.10 RouteCollection.Remove Method

Overload List

	Name	Description
	Remove(String) [▶ 1303]	Removes the specified route from the RouteCollection [▶ 1292].
	Remove(AmsNetId) [▶ 1304]	Removes the specified AmsNetId [▶ 665] from the RouteCollection [▶ 1292].
	Remove(Route) [▶ 1304]	Removes the first occurrence of a specific object from the ICollection.T..

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

RouteCollection.Remove Method (String)

Removes the specified route from the RouteCollection [▶ 1292].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    string routeName
)
```

Parameters

routeName	Type: System.String Name of the route.

Return Value

Type: Boolean
true if removed, false otherwise.

Reference

RouteCollection Class [▶ 1292]

Remove Overload [▶ 1303]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

RouteCollection.Remove Method (AmsNetId)

Removes the specified AmsNetId [▶ 665] from the RouteCollection [▶ 1292].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    AmsNetId netId
)
```

Parameters

netId	Type: TwinCAT.Ads.AmsNetId [▶ 665] The Ams Net ID..

Return Value

Type: Boolean
true if removed, false otherwise.

Reference

RouteCollection Class [▶ 1292]

Remove Overload [▶ 1303]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

RouteCollection.Remove Method (Route)

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    Route item
)
```

Parameters

item	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The object to remove from the ICollection.T..

Return Value

Type: Boolean
. if item was successfully removed from the ICollection.T.; otherwise, .. This method also returns . if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

RouteCollection Class [▶ 1292]

Remove Overload [▶ 1303]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.11 RouteCollection.RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.4.3.12 RouteCollection.TryAdd Method

Tries to add the route to the RouteCollection [▶ 1292].

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryAdd(
    Route route
)
```

Parameters

route	Type: TwinCAT.Ads.TcpRouter.Route [▶ 1281] The route.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

RouteCollection Class [▶ 1292]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5 RouterException Class

An RouterException is thrown on communication errors in the AmsTcpIpRouter [▶ 1254] class.

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.TcpRouter.RouterException
      TwinCAT.Ads.TcpRouter.RouterNotInitializedException [▶ 1310]
      TwinCAT.Ads.TcpRouter.RouterNotStartedException [▶ 1313]

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class RouterException : Exception
```

The RouterException type exposes the following members.

Constructors

	Name	Description
	RouterException [▶ 1308]	Initializes a new instance of the RouterException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5.1 RouterException Constructor

Initializes a new instance of the RouterException [▶ 1306] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected RouterException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

RouterException Class [▶ 1306]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5.2 RouterException Properties

The RouterException [▶ 1306] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

RouterException Class [▶ 1306]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5.3 RouterException Methods

The RouterException [▶ 1306] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

RouterException Class [▶ 1306]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5.3.1 RouterException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

RouterException Class [▶ 1306]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.5.4 RouterException Events

The RouterException [▶ 1306] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

RouterException Class [▶ 1306]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.6 RouterNotInitializedException Class

Class RouterNotInitializedException. Implements the RouterException [▶ 1306]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.TcpRouter.RouterException [▶ 1306]
      TwinCAT.Ads.TcpRouter.RouterNotInitializedException

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RouterNotInitializedException : RouterException
```

The RouterNotInitializedException type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [▶ 1306].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

TwinCAT.Ads.TcpRouter.RouterException [▶ 1306]

6.6.6.1 RouterNotInitializedException Properties

The RouterNotInitializedException [▶ 1310] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

RouterNotInitializedException Class [▶ 1310]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.6.2 RouterNotInitializedException Methods

The RouterNotInitializedException [▶ 1310] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [▶ 1306].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

RouterNotInitializedException Class [▶ 1310]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.6.3 RouterNotInitializedException Events

The RouterNotInitializedException [▶ 1310] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

RouterNotInitializedException Class [▶ 1310]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.7 RouterNotStartedException Class

Class RouterNotStartedException. Implements the RouterException [▶ 1306]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ads.TcpRouter.RouterException [▶ 1306]
      TwinCAT.Ads.TcpRouter.RouterNotStartedException

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RouterNotStartedException : RouterException
```

The RouterNotStartedException type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [▶ 1306].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

TwinCAT.Ads.TcpRouter.RouterException [▶ 1306]

6.6.7.1 RouterNotStartedException Properties

The RouterNotStartedException [▶ 1313] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

RouterNotStartedException Class [▶ 1313]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.7.2 RouterNotStartedException Methods

The RouterNotStartedException [▶ 1313] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1309]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from RouterException [▶ 1306].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

RouterNotStartedException Class [▶ 1313]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.7.3 RouterNotStartedException Events

The RouterNotStartedException [▶ 1313] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

RouterNotStartedException Class [▶ 1313]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.8 RouterStatus Enumeration

Enum RouterStatus

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum RouterStatus
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Initializing	1	Initializing
	Starting	2	Router Starting
	Started	3	Router Started / Executing
	Stopping	4	Router Stopping.
	Stopped	5	Router stopped.

Remarks

Indicates the status of the router.

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.9 RouterStatusChangedEventArgs Class

Event Arguments sent when the RouterStatusChanged [▶ 1281]. Implements the EventArgs

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.Ads.TcpRouter.RouterStatusChangedEventArgs
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RouterStatusChangedEventArgs : EventArgs
```

The RouterStatusChangedEventArgs type exposes the following members.

Constructors

	Name	Description
	RouterStatusChangedEventArgs [▶ 1318]	Initializes a new instance of the RouterStatusChangedEventArgs class.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Fields

	Name	Description
	RouterStatus [▶ 1319]	The router status

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

System.EventArgs

IAmsRouter.RouterStatusChanged [▶ 1281]

6.6.9.1 RouterStatusChangedEventArgs Constructor

Initializes a new instance of the RouterStatusChangedEventArgs [▶ 1317] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RouterStatusChangedEventArgs(
    RouterStatus status
)
```

Parameters

status	Type: TwinCAT.Ads.TcpRouter.RouterStatus [▶ 1316] The status.

Reference

RouterStatusChangedEventArgs Class [▶ 1317]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.9.2 RouterStatusChangedEventArgs Methods

The RouterStatusChangedEventArgs [▶ 1317] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

RouterStatusChangedEventArgs Class [▶ 1317]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.9.3 RouterStatusChangedEventArgs Fields

The RouterStatusChangedEventArgs [▶ 1317] type exposes the following members.

Fields

	Name	Description
	RouterStatus [▶ 1319]	The router status

Reference

RouterStatusChangedEventArgs Class [▶ 1317]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.9.3.1 RouterStatusChangedEventArgs.RouterStatus Field

The router status

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public readonly RouterStatus RouterStatus
```

Field Value

Type: RouterStatus [▶ 1316]

Reference

RouterStatusChangedEventArgs Class [▶ 1317]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.10 StaticRoutesXmlConfigurationBuilderExtension Class

Extension class adding StaticRoutes.xml file reading to the IConfigurationBuilder.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationBuilderExtension
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class StaticRoutesXmlConfigurationBuilderExtension
```

The StaticRoutesXmlConfigurationBuilderExtension type exposes the following members.

Methods

	Name	Description
	AddStaticRoutesXmlConfiguration [▶ 1320]	Adds the static routes XML configuration.

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.10.1 StaticRoutesXmlConfigurationBuilderExtension Methods

The StaticRoutesXmlConfigurationBuilderExtension [▶ 1320] type exposes the following members.

Methods

	Name	Description
	AddStaticRoutesXmlConfiguration [▶ 1320]	Adds the static routes XML configuration.

Reference

StaticRoutesXmlConfigurationBuilderExtension Class [▶ 1320]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.10.1.1 StaticRoutesXmlConfigurationBuilderExtension.AddStaticRoutesXmlConfiguration Method

Adds the static routes XML configuration.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IConfigurationBuilder AddStaticRoutesXmlConfiguration(
    this IConfigurationBuilder builder
)
```

Parameters

builder	Type: IConfigurationBuilder The builder.

Return Value

Type: IConfigurationBuilder
IConfigurationBuilder.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type IConfigurationBuilder. When you use instance method syntax to call this method, omit the first parameter. For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

Reference

StaticRoutesXmlConfigurationBuilderExtension Class [▶ 1320]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11 StaticRoutesXmlConfigurationProvider Class

Class StaticRoutesXmlConfigurationProvider. Implements the ConfigurationProvider

Inheritance Hierarchy

System.Object
  ConfigurationProvider
    TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationProvider
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class StaticRoutesXmlConfigurationProvider : ConfigurationProvider
```

The StaticRoutesXmlConfigurationProvider type exposes the following members.

Constructors

	Name	Description
	StaticRoutesXmlConfigurationProvider [▶ 1322]	Initializes a new instance of the StaticRoutesXmlConfigurationProvider class.

Properties

	Name	Description
	Data	The configuration key value pairs for this provider. (Inherited from ConfigurationProvider.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetChildKeys [▶ 1324]	Returns the list of keys that this provider has. (Overrides ConfigurationProvider.GetChildKeys(IEnumerable.String., String).)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetReloadToken	Returns a IChangeToken that can be used to listen when this provider is reloaded. (Inherited from ConfigurationProvider.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Load [▶ 1324]	Loads (or reloads) the data for this provider. (Overrides ConfigurationProvider.Load..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnReload	Triggers the reload change token and creates a new one. (Inherited from ConfigurationProvider.)
	Set	Sets a value for a given key. (Inherited from ConfigurationProvider.)
	ToString	Generates a string representing this provider name and relevant details. (Inherited from ConfigurationProvider.)
	TryGet [▶ 1325]	Attempts to find a value with the given key, returns true if one is found, false otherwise. (Overrides ConfigurationProvider.TryGet(String, String.).)

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

ConfigurationProvider

6.6.11.1 StaticRoutesXmlConfigurationProvider Constructor

Initializes a new instance of the StaticRoutesXmlConfigurationProvider [▶ 1321] class.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StaticRoutesXmlConfigurationProvider()
```

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11.2 StaticRoutesXmlConfigurationProvider Properties

The StaticRoutesXmlConfigurationProvider [▶ 1321] type exposes the following members.

Properties

	Name	Description
	Data	The configuration key value pairs for this provider. (Inherited from ConfigurationProvider.)

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11.3 StaticRoutesXmlConfigurationProvider Methods

The StaticRoutesXmlConfigurationProvider [▶ 1321] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetChildKeys [▶ 1324]	Returns the list of keys that this provider has. (Overrides ConfigurationProvider.GetChildKeys(IEnumerable.String., String).)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetReloadToken	Returns a IChangeToken that can be used to listen when this provider is reloaded. (Inherited from ConfigurationProvider.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Load [▶ 1324]	Loads (or reloads) the data for this provider. (Overrides ConfigurationProvider.Load..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnReload	Triggers the reload change token and creates a new one. (Inherited from ConfigurationProvider.)
	Set	Sets a value for a given key. (Inherited from ConfigurationProvider.)

	Name	Description
	ToString	Generates a string representing this provider name and relevant details. (Inherited from ConfigurationProvider.)
	TryGet [▶ 1325]	Attempts to find a value with the given key, returns true if one is found, false otherwise. (Overrides ConfigurationProvider.TryGet(String, String).)

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11.3.1 StaticRoutesXmlConfigurationProvider.GetChildKeys Method

Returns the list of keys that this provider has.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetChildKeys(
    IEnumerable<string> earlierKeys,
    string parentPath
)
```

Parameters

earlierKeys	Type: System.Collections.Generic.IEnumerable.String. The earlier keys that other providers contain.
parentPath	Type: System.String The path for the parent IConfiguration.

Return Value

Type: IEnumerable.String.
The list of keys for this provider.

Implements

IConfigurationProvider.GetChildKeys(IEnumerable.String., String)

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11.3.2 StaticRoutesXmlConfigurationProvider.Load Method

Loads (or reloads) the data for this provider.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void Load()
```

Implements

IConfigurationProvider.Load.

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.11.3.3 StaticRoutesXmlConfigurationProvider.TryGet Method

Attempts to find a value with the given key, returns true if one is found, false otherwise.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGet(
    string key,
    out string value
)
```

Parameters

key	Type: System.String The key to lookup.
value	Type: System.String. The value found at key if one is found.

Return Value

Type: Boolean
True if key has a value, false otherwise.

Implements

IConfigurationProvider.TryGet(String, String.)

Reference

StaticRoutesXmlConfigurationProvider Class [▶ 1321]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.12 StaticRoutesXmlConfigurationSource Class

StaticRoutes Configuration Sources Implements the IConfigurationSource

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TcpRouter.StaticRoutesXmlConfigurationSource
Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class StaticRoutesXmlConfigurationSource : IConfigurationSource
```

The StaticRoutesXmlConfigurationSource type exposes the following members.

Constructors

	Name	Description
	StaticRoutesXmlConfigurationSource [▶ 1326]	

Methods

	Name	Description
	Build [▶ 1327]	Builds the IConfigurationProvider for this source.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

IConfigurationSource

6.6.12.1 StaticRoutesXmlConfigurationSource Constructor

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public StaticRoutesXmlConfigurationSource()
```

Reference

StaticRoutesXmlConfigurationSource Class [▶ 1326]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.12.2 StaticRoutesXmlConfigurationSource Methods

The StaticRoutesXmlConfigurationSource [▶ 1326] type exposes the following members.

Methods

	Name	Description
	Build [▶ 1327]	Builds the IConfigurationProvider for this source.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

StaticRoutesXmlConfigurationSource Class [▶ 1326]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.6.12.2.1 StaticRoutesXmlConfigurationSource.Build Method

Builds the IConfigurationProvider for this source.

Namespace: TwinCAT.Ads.TcpRouter [▶ 1254]
Assembly: TwinCAT.Ads.TcpRouter (in TwinCAT.Ads.TcpRouter.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConfigurationProvider Build(
    IConfigurationBuilder builder
)
```

Parameters

builder	Type: IConfigurationBuilder The IConfigurationBuilder.

Return Value

Type: IConfigurationProvider
An IConfigurationProvider

Implements

IConfigurationSource.Build(IConfigurationBuilder)

Reference

StaticRoutesXmlConfigurationSource Class [▶ 1326]

TwinCAT.Ads.TcpRouter Namespace [▶ 1254]

6.7 TwinCAT.Ads.TypeSystem Namespace

Root namespace for the ADS type system.

Classes

	Class	Description
	AliasType [▶ 1329]	Alias DataType
	ArrayType [▶ 1335]	Represents an Array DataType
	BitMappingType [▶ 1345]	Helper Data Type to implement Bit mapping types.
	DataType [▶ 1349]	DataType class
	EnumType.T. [▶ 1363]	Enum DataType [▶ 1349].
	Field [▶ 1374]	Represents a field of an Struct/Alias/Union
	Instance [▶ 1388]	Instance implementation
	Member [▶ 1406]	Represents a member of an StructType [▶ 1457]
	PCCHType [▶ 1411]	Class PCCHType. This class cannot be inherited. Implements the PointerType [▶ 1414]
	PointerType [▶ 1414]	Represents a pointer type.
	PrimitiveType [▶ 1418]	Class PrimitiveType.
	PVoidType [▶ 1422]	Class PVoidType. This class cannot be inherited. Implements the PointerType [▶ 1414]
	ReferenceType [▶ 1425]	Represents a reference type
	RpcMethod [▶ 1433]	RPC Method Description

	Class	Description
	RpcMethodParameter [▶ 1440]	Class RpcMethodParameter.
	RpcStructType [▶ 1447]	StructType which is callable by RPC Methods.
	StringType [▶ 1452]	String DataType
	StructType [▶ 1457]	Represents a struct type
	SubRangeType.T. [▶ 1463]	Represents a SubRangType
	Symbol [▶ 1469]	Symbol class
	SymbolIterator [▶ 1517]	Iterator class for enumerations of Symbols [▶ 2176].
	SymbolLoaderFactory [▶ 1523]	The class SymbolLoaderFactory [▶ 1523] is used to create a new instance of the AdsSymbolLoader initialized to the parametrized mode (SymbolBrowser V2, new Version)
	UnionType [▶ 1533]	Represents a union type
	WStringType [▶ 1537]	Represents an Unicode string (Wide string)

Interfaces

	Interface	Description
	IAdsSymbol [▶ 1379]	Interface IAdsSymbol
	IAdsSymbolLoader [▶ 1383]	Symbol Loader interface
	IContextMaskProvider [▶ 1387]	Interface IContextMaskProvider

6.7.1 AliasType Class

Alias DataType

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.AliasType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class AliasType : DataType,
    IAliasType, IDataType, IBitSize
```

The AliasType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1332]	Gets the Base Type
	BaseTypeName [▶ 1332]	Gets the BaseType name
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1333]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Overrides DataType.IsContainer [▶ 1357].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1334]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1334]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	ToString [▶ 1335]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.1 AliasType Properties

The AliasType [▶ 1329] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1332]	Gets the Base Type
	BaseTypeName [▶ 1332]	Gets the BaseType name
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1333]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Overrides DataType.IsContainer [▶ 1357].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1334]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1334]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)

	Name	Description
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.1.1 AliasType.BaseType Property

Gets the Base Type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]

Implements

IAliasType.BaseType [▶ 1956]

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.1.2 AliasType.BaseTypeName Property

Gets the BaseType name

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string BaseTypeName { get; }
```

Property Value

Type: String

Implements

IAliasType.BaseTypeName [▶ 1957]

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.1.3 AliasType.IsContainer Property

Gets a value indicating whether this IDataType [▶ 1986] is a container type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool IsContainer { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

IDataType.IsContainer [▶ 1990]
IDataType.IsContainer [▶ 1990]

Remarks

Container Types are all types that contain SubElements like

- Array [▶ 1649]
- Pointer [▶ 1649]
- Union [▶ 1649]
- Struct [▶ 1649]
- Function [▶ 1649]
- FunctionBlock [▶ 1649]
- Program [▶ 1649]

And the Alias [▶ 1649] types, if they have a container type as base type.

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.1.1.4 AliasType.IsPrimitive Property

Gets a value indicating whether this IDataType [▶ 1986] is primitive

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive [▶ 1991]
IDataType.IsPrimitive [▶ 1991]

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.1.5 AliasType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override Type ManagedType { get; }
```

Property Value

Type: Type
Dot net type.

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.2 AliasType Methods

The AliasType [▶ 1329] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1335]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.1.2.1 AliasType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

AliasType Class [▶ 1329]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2 ArrayType Class

Represents an Array DataType

DataType class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.ArrayType
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ArrayType : DataType, IArrayType,
    IDataType, IBitSize
```

The ArrayType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	DimensionCount [▶ 1339]	Gets the dimension count.
	Dimensions [▶ 1339]	Gets the dimensions as read only collection.
	ElementCount [▶ 1339]	Gets the element count.
	ElementSize [▶ 1340]	Gets the byte-size of a single element of the array
	ElementType [▶ 1340]	Gets the type of the contained elements.
	ElementTypeName [▶ 1341]	Gets the name of the element type.
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsJagged [▶ 1341]	Gets a value indicating whether this instance is jagged.
	IsOversampled [▶ 1342]	Gets a value indicating whether this array instance describes an oversampling type.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1342]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)

	Name	Description
	JaggedLevel [▶ 1343]	Gets the jagged level (Non-Jagged Array have level 1)
	ManagedType [▶ 1343]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetDimensions [▶ 1344]	Sets the dimensions.
	SetElementType [▶ 1345]	Sets the type of the element.
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1 ArrayType Properties

The ArrayType [▶ 1335] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)

	Name	Description
	DimensionCount [▶ 1339]	Gets the dimension count.
	Dimensions [▶ 1339]	Gets the dimensions as read only collection.
	ElementCount [▶ 1339]	Gets the element count.
	ElementSize [▶ 1340]	Gets the byte-size of a single element of the array
	ElementType [▶ 1340]	Gets the type of the contained elements.
	ElementTypeName [▶ 1341]	Gets the name of the element type.
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsJagged [▶ 1341]	Gets a value indicating whether this instance is jagged.
	IsOversampled [▶ 1342]	Gets a value indicating whether this array instance describes an oversampling type.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1342]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	JaggedLevel [▶ 1343]	Gets the jagged level (Non-Jagged Array have level 1)
	ManagedType [▶ 1343]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.1 ArrayType.DimensionCount Property

Gets the dimension count.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int DimensionCount { get; }
```

Property Value

Type: Int32
The dimension count.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.2 ArrayType.Dimensions Property

Gets the dimensions as read only collection.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDimensionCollection Dimensions { get; }
```

Property Value

Type: IDimensionCollection [▶ 2000]
The dimensions.

Implements

IArrayType.Dimensions [▶ 1974]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.3 ArrayType.ElementCount Property

Gets the element count.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ElementCount { get; }
```

Property Value

Type: Int32
The element count.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.4 ArrayType.ElementSize Property

Gets the byte-size of a single element of the array

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ElementSize { get; }
```

Property Value

Type: Int32
The size of the element.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.5 ArrayType.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ElementType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the element.

Implements

IArrayType.ElementType [▶ 1974]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.6 ArrayType.ElementTypeName Property

Gets the name of the element type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ElementTypeName { get; }
```

Property Value

Type: String
The name of the element type.

Implements

IArrayType.ElementTypeName [▶ 1974]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.7 ArrayType.IsJagged Property

Gets a value indicating whether this instance is jagged.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsJagged { get; }
```

Property Value

Type: Boolean
true if this instance is jagged; otherwise, false.

Implements

IArrayType.IsJagged [▶ 1975]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.8 ArrayType.IsOversampled Property

Gets a value indicating whether this array instance describes an oversampling type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsOversampled { get; }
```

Property Value

Type: Boolean
true if this instance is oversampling; otherwise, false.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.9 ArrayType.IsPrimitive Property

Gets a value indicating whether this IDataType [▶ 1986] is primitive

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive [▶ 1991]
IDataType.IsPrimitive [▶ 1991]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.10 ArrayType.JaggedLevel Property

Gets the jagged level (Non-Jagged Array have level 1)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int JaggedLevel { get; }
```

Property Value

Type: Int32
The jagged level.

Implements

IArrayType.JaggedLevel [▶ 1975]

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.1.11 ArrayType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override Type ManagedType { get; }
```

Property Value

Type: Type
Dot net type.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.2 ArrayType Methods

The ArrayType [▶ 1335] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetDimensions [▶ 1344]	Sets the dimensions.
	SetElementType [▶ 1345]	Sets the type of the element.
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.2.1 ArrayType.SetDimensions Method

Sets the dimensions.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetDimensions(
    DimensionCollection dims
)
```

Parameters

dims	Type: TwinCAT.TypeSystem.DimensionCollection [▶ 1671] The dims.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.2.2.2 ArrayType.SetElementType Method

Sets the type of the element.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetElementType(
    DataType elementType
)
```

Parameters

elementType	Type: TwinCAT.Ads.TypeSystem.DataType [▶ 1349] Type of the element.

Reference

ArrayType Class [▶ 1335]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.3 BitMappingType Class

Helper Data Type to implement Bit mapping types.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.BitMappingType
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class BitMappingType : DataType
```

The BitMappingType type exposes the following members.

Constructors

	Name	Description
	BitMappingType [▶ 1347]	Initializes a new instance of the BitMappingType class.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.3.1 BitMappingType Constructor

Initializes a new instance of the BitMappingType [▶ 1345] class.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public BitMappingType(
    string name,
    int bitSize,
    Type dotnetType
)
```

Parameters

name	Type: System.String The name.
bitSize	Type: System.Int32 The size of the type in bits.
dotnetType	Type: System.Type Type of the dotnet.

Reference

BitMappingType Class [▶ 1345]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.3.2 BitMappingType Properties

The BitMappingType [▶ 1345] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)

	Name	Description
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

BitMappingType Class [▶ 1345]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.3.3 BitMappingType Methods

The BitMappingType [▶ 1345] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

BitMappingType Class [▶ 1345]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4 DataType Class

DataType class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType
    More... [▶ 1350]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DataType : IDataType, IBitSize
```

The DataType type exposes the following members.

Constructors

	Name	Description
	DataType [▶ 1351]	Initializes a new instance of the DataType class (copy Constructor)

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986]
	BitSize [▶ 1353]	Gets the size of the DataType in bits.
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1354]	Gets the Data Type category
	Comment [▶ 1354]	Gets the comment.
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name)
	Id [▶ 1355]	Gets the ID of the DataType
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive

	Name	Description
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached.
	Name [▶ 1360]	Gets the name of the Data Type (without namespace)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists.
	Size [▶ 1361]	Gets the Size of the DataType in Bytes or bits.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	IsPointerType [▶ 1362]	Determines whether the specified category is a pointer type.
S	IsReferenceType [▶ 1362]	Determines whether the specified category is a reference type.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType
    TwinCAT.Ads.TypeSystem.AliasType [▶ 1329]
    TwinCAT.Ads.TypeSystem.ArrayType [▶ 1335]
    TwinCAT.Ads.TypeSystem.BitMappingType [▶ 1345]
    TwinCAT.Ads.TypeSystem.EnumType.T. [▶ 1363]
    TwinCAT.Ads.TypeSystem.PointerType [▶ 1414]
    TwinCAT.Ads.TypeSystem.PrimitiveType [▶ 1418]
    TwinCAT.Ads.TypeSystem.ReferenceType [▶ 1425]
    TwinCAT.Ads.TypeSystem.StringType [▶ 1452]
    TwinCAT.Ads.TypeSystem.StructType [▶ 1457]
    TwinCAT.Ads.TypeSystem.SubRangeType.T. [▶ 1463]
    TwinCAT.Ads.TypeSystem.UnionType [▶ 1533]
    TwinCAT.Ads.TypeSystem.WStringType [▶ 1537]

6.7.4.1 DataType Constructor

Initializes a new instance of the DataType [▶ 1349] class (copy Constructor)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DataType(
    DataType copy
)
```

Parameters

copy	Type: TwinCAT.Ads.TypeSystem.DataType [▶ 1349] The copy.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2 DataType Properties

The DataType [▶ 1349] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986]
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits.
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1354]	Gets the Data Type category
	Comment [▶ 1354]	Gets the comment.
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name)
	Id [▶ 1355]	Gets the ID of the DataType
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive

	Name	Description
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached.
	Name [▶ 1360]	Gets the name of the Data Type (without namespace)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists.
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.1 DataType.Attributes Property

Gets the attributes of the IDataType [▶ 1986]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection [▶ 2211]
The attributes.

Implements

IDataType.Attributes [▶ 1987]

Exceptions

Exception	Condition
NotImplementedException	

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.2 DataType.BitSize Property

Gets the size of the DataType [▶ 1349] in bits.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int BitSize { get; }
```

Property Value

Type: Int32
The size of the bit.

Implements

IBitSize.BitSize [▶ 1984]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.3 DataType.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ByteSize { get; }
```

Property Value

Type: Int32
The size of the byte.

Implements

IBitSize.ByteSize [▶ 1984]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.4 DataType.Category Property

Gets the Data Type category

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory Category { get; protected set; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The category.

Implements

IDataType.Category [▶ 1988]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.5 DataType.Comment Property

Gets the comment.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Comment { get; }
```

Property Value

Type: String
The comment.

Implements

IDataType.Comment [▶ 1988]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.6 DataType.FullName Property

Gets the full name of the IDataType [▶ 1986] (Namespace + Name)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string FullName { get; }
```

Property Value

Type: String
The full name.

Implements

IDataType.FullName [▶ 1989]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.7 DataType.Id Property

Gets the ID of the DataType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Id { get; }
```

Property Value

Type: Int32
The id.

Implements

IDataType.Id [▶ 1989]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.8 DataType.IsBitType Property

Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping subtype; otherwise, false.

Implements

IBitSize.IsBitType [▶ 1984]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.9 DataType.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Implements

IBitSize.IsByteAligned [▶ 1985]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.10 DataType.IsContainer Property

Gets a value indicating whether this IDataType [▶ 1986] is a container type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsContainer { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

IDataType.IsContainer [▶ 1990]

Remarks

Container Types are all types that contain SubElements like

- Array [▶ 1649]
- Pointer [▶ 1649]
- Union [▶ 1649]
- Struct [▶ 1649]
- Function [▶ 1649]
- FunctionBlock [▶ 1649]
- Program [▶ 1649]

and the Alias [▶ 1649] and Reference [▶ 1649] types, if they have a container type as base type.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.4.2.11 DataType.IsPointer Property

Gets a value indicating whether this IDataType [▶ 1986] is a pointer type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsPointer { get; }
```

Property Value

Type: Boolean
true if this instance is pointer type; otherwise, false.

Implements

IDataType.IsPointer [▶ 1990]

Remarks

Pointer types can be dereferenced with the '^' operator.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.4.2.12 DataType.IsPrimitive Property

Gets a value indicating whether this IDataType [▶ 1986] is primitive

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive [▶ 1991]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.13 DataType.IsReference Property

Gets a value indicating whether this IDataType [▶ 1986] is a reference type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsReference { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

IDataType.IsReference [▶ 1991]

Remarks

Reference types can be dereferenced.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.4.2.14 DataType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual Type ManagedType { get; protected set; }
```

Property Value

Type: Type
Dot net type.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.15 DataType.Name Property

Gets the name of the Data Type (without namespace)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; protected set; }
```

Property Value

Type: String
The name.

Implements

IDataType.Name [▶ 1992]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.16 DataType.Namespace Property

Gets the namespace string within the IDataType [▶ 1986] exists.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Namespace { get; }
```

Property Value

Type: String
The namespace.

Implements

IDataType.Namespace [▶ 1992]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.2.17 DataType.Size Property

Gets the Size of the DataType [▶ 1349] in Bytes or bits.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Size { get; protected set; }
```

Property Value

Type: Int32
The size.

Implements

IBitSize.Size [▶ 1985]

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.3 DataType Methods

The DataType [▶ 1349] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	IsPointerType [▶ 1362]	Determines whether the specified category is a pointer type.
S	IsReferenceType [▶ 1362]	Determines whether the specified category is a reference type.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.3.1 DataType.IsPointerType Method

Determines whether the specified category is a pointer type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool IsPointerType(
    DataTypeCategory cat
)
```

Parameters

cat	Type: TwinCAT.TypeSystem.DataTypeCategory [▶ 1649] The data type category.

Return Value

Type: Boolean
true if [is pointer type] [the specified cat]; otherwise, false.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.3.2 DataType.IsReferenceType Method

Determines whether the specified category is a reference type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool IsReferenceType(
    DataTypeCategory cat
)
```

Parameters

cat	Type: TwinCAT.TypeSystem.DataTypeCategory [▶ 1649] The data type category.

Return Value

Type: Boolean
true if [is reference type] [the specified cat]; otherwise, false.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.4.3.3 DataType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DataType Class [▶ 1349]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5 EnumType.T. Class

Enum DataType [▶ 1349].

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.EnumType.T.
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class EnumType<T> : DataType,
    IEnumType<T>, IAliasType, IDataType, IBitSize, IEnumType
where T : IConvertible
```

Type Parameters

T

The EnumType.T. type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1366]	Gets the Base Type
	BaseTypeName [▶ 1367]	Gets the BaseType name
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	EnumValues [▶ 1367]	Enumeration specification (if enum)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Contains [▶ 1368]	Determines whether the enum values contains the specified name
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1369]	Gets the filed names of the IEnumType.T. [▶ 2021]
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1370]	Gets the values of the IEnumType.T. [▶ 2021]
	Parse [▶ 1370]	Parses a name of the IEnumType.T. [▶ 2021] and returns the value (as base type)
	ToString. [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)
	ToString(IConvertible) [▶ 1371]	Returns a String that represents this instance.
	ToString(T) [▶ 1372]	Returns a String that represents this instance.
	TryParse(String, IEnumValue.) [▶ 1373]	Tries to parse the Enum Value
	TryParse(String, T.) [▶ 1374]	Tries to parse the Enum Value

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.1 EnumType.T. Properties

The EnumType.T. [▶ 1363] generic type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1366]	Gets the Base Type
	BaseTypeName [▶ 1367]	Gets the BaseType name
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)

	Name	Description
	EnumValues [▶ 1367]	Enumeration specification (if enum)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.1.1 EnumType.T..BaseType Property

Gets the Base Type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the base.

Implements

IAliasType.BaseType [▶ 1956]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.1.2 EnumType.T..BaseTypeName Property

Gets the BaseType name

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string BaseTypeName { get; }
```

Property Value

Type: String
The name of the base type.

Implements

IAliasType.BaseTypeName [▶ 1957]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.1.3 EnumType.T..EnumValues Property

Enumeration specification (if enum)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumValueCollection EnumValues { get; }
```

Property Value

Type: IEnumValueCollection [▶ 2031]
The enum specification.

Implements

IEnumType.T..EnumValues [▶ 2024]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2 EnumType.T. Methods

The EnumType.T. [▶ 1363] generic type exposes the following members.

Methods

	Name	Description
	Contains [▶ 1368]	Determines whether the enum values contains the specified name
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1369]	Gets the filed names of the IEnumType.T. [▶ 2021]
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1370]	Gets the values of the IEnumType.T. [▶ 2021]
	Parse [▶ 1370]	Parses a name of the IEnumType.T. [▶ 2021] and returns the value (as base type)
	ToString. [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)
	ToString(IConvertible) [▶ 1371]	Returns a String that represents this instance.
	ToString(T) [▶ 1372]	Returns a String that represents this instance.
	TryParse(String, IEnumValue.) [▶ 1373]	Tries to parse the Enum Value
	TryParse(String, T.) [▶ 1374]	Tries to parse the Enum Value

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.1 EnumType.T..Contains Method

Determines whether the enum values contains the specified name

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if contains the value, otherwise, false.

Implements

IEnumType.T..Contains(String) [▶ 2025]
IEnumType.Contains(String) [▶ 2017]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.2 EnumType.T..GetNames Method

Gets the filed names of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Implements

IEnumType.T..GetNames. [▶ 2025]
IEnumType.GetNames. [▶ 2018]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.3 EnumType.T..GetValues Method

Gets the values of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T[] GetValues()
```

Return Value

Type: .T [▶ 1363].
T[].

Implements

IEnumType.T..GetValues. [▶ 2026]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.4 EnumType.T..Parse Method

Parses a name of the IEnumType.T. [▶ 2021] and returns the value (as base type)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Parse(
    string strValue
)
```

Parameters

strValue	Type: System.String Enum Value as string.

Return Value

Type: T [▶ 1363]
T.

Implements

IEnumType.T..Parse(String) [▶ 2026]

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.5 EnumType.T..ToString Method

Overload List

	Name	Description
	ToString. [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)
	ToString(IConvertible) [▶ 1371]	Returns a String that represents this instance.
	ToString(T) [▶ 1372]	Returns a String that represents this instance.

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

EnumType.T..ToString Method (IConvertible)

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ToString(
    IConvertible val
)
```

Parameters

val	Type: System.IConvertible The value.

Return Value

Type: String
A String that represents this instance.

Implements

IEnumType.ToString(IConvertible) [▶ 2019]

Reference

EnumType.T. Class [▶ 1363]

ToString Overload [▶ 1371]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

EnumType.T..ToString Method (T)

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ToString(
    T val
)
```

Parameters

val	Type: T [▶ 1363] The value.

Return Value

Type: String
A String that represents this instance.

Implements

IEnumType.T..ToString(T) [▶ 2027]

Exceptions

Exception	Condition
ArgumentException	val

Reference

EnumType.T. Class [▶ 1363]

ToString Overload [▶ 1371]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.5.2.6 EnumType.T..TryParse Method

Overload List

	Name	Description
	TryParse(String, IEnumValue.) [▶ 1373]	Tries to parse the Enum Value
	TryParse(String, T.) [▶ 1374]	Tries to parse the Enum Value

Reference

EnumType.T. Class [▶ 1363]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

EnumType.T..TryParse Method (String, IEnumValue.)

Tries to parse the Enum Value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out IEnumValue value
)
```

Parameters

strValue	Type: System.String Enum value (in string representation).
value	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumType.TryParse(String, IEnumValue.) [▶ 2021]

Reference

EnumType.T. Class [▶ 1363]

TryParse Overload [▶ 1373]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

EnumType.T..TryParse Method (String, T.)

Tries to parse the Enum Value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out T value
)
```

Parameters

strValue	Type: System.String Enum value (in string representation).
value	Type: T [▶ 1363]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumType.T..TryParse(String, T.) [▶ 2027]

Reference

EnumType.T. Class [▶ 1363]

TryParse Overload [▶ 1373]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.6 Field Class

Represents a field of an Struct/Alias/Union

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.Instance [▶ 1388]
    TwinCAT.Ads.TypeSystem.Field
      TwinCAT.Ads.TypeSystem.Member [▶ 1406]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Field : Instance, IField,
    IAttributedInstance, IInstance, IBitSize
```

The Field type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance [▶ 1388].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)

	Name	Description
	ParentType [▶ 1378]	Gets the Parent of this IField [▶ 2040].
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1378]	Gets the value encoding of this Field

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance [▶ 1388] (Inherited from Instance [▶ 1388].)
	OnSetInstanceName [▶ 1404]	Sets a new InstanceName InstancePath (Inherited from Instance [▶ 1388].)
	SetAttributes [▶ 1404]	Sets the type attributes (Inherited from Instance [▶ 1388].)
	SetContextMask [▶ 1405]	Sets the context mask. (Inherited from Instance [▶ 1388].)
	ToString [▶ 1405]	Returns a String that represents this instance. (Inherited from Instance [▶ 1388].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.6.1 Field Properties

The Field [▶ 1374] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)

	Name	Description
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance [▶ 1388].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)
	ParentType [▶ 1378]	Gets the Parent of this IField [▶ 2040].
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1378]	Gets the value encoding of this Field [▶ 1374]

Reference

Field Class [▶ 1374]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.6.1.1 Field.ParentType Property

Gets the Parent of this IField [▶ 2040].

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ParentType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the parent (Alias, Union, Struct)

Implements

IField.ParentType [▶ 2042]

Reference

Field Class [▶ 1374]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.6.1.2 Field.ValueEncoding Property

Gets the value encoding of this Field [▶ 1374]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual Encoding ValueEncoding { get; }
```

Property Value

Type: Encoding
The value encoding.

Implements

IAttributedInstance.ValueEncoding [▶ 1982]

Reference

Field Class [▶ 1374]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.6.2 Field Methods

The Field [▶ 1374] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance [▶ 1388] (Inherited from Instance [▶ 1388].)
	OnSetInstanceName [▶ 1404]	Sets a new InstanceName InstancePath (Inherited from Instance [▶ 1388].)
	SetAttributes [▶ 1404]	Sets the type attributes (Inherited from Instance [▶ 1388].)
	SetContextMask [▶ 1405]	Sets the context mask. (Inherited from Instance [▶ 1388].)
	ToString [▶ 1405]	Returns a String that represents this instance. (Inherited from Instance [▶ 1388].)

Reference

Field Class [▶ 1374]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.7 IAdsSymbol Interface

Interface IAdsSymbol

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize, IProcessImageAddress, IContextMaskProvider
```

The IAdsSymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	ContextMask [▶ 1388]	Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications. (Inherited from IContextMaskProvider [▶ 1387].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	DataTypeId [▶ 1382]	DataType identifier of the Symbol AdsDataTypeId [▶ 574]
	ImageBaseAddress [▶ 1383]	Gets the AmsAddress [▶ 648] of the Process Image
	IndexGroup [▶ 2093]	Gets the index group of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)
	IndexOffset [▶ 2093]	Gets the index offset of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	IsVirtual [▶ 2094]	Gets a value indicating whether this instance is virtual. (Inherited from IProcessImageAddress [▶ 2092].)

	Name	Description
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

TwinCAT.TypeSystem.IProcessImageAddress [▶ 2092]

6.7.7.1 IAdsSymbol Properties

The IAdsSymbol [▶ 1379] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	ContextMask [▶ 1388]	Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications. (Inherited from IContextMaskProvider [▶ 1387].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	DataTypeId [▶ 1382]	DataType identifier of the Symbol AdsDataTypeId [▶ 574]
	ImageBaseAddress [▶ 1383]	Gets the AmsAddress [▶ 648] of the Process Image
	IndexGroup [▶ 2093]	Gets the index group of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)
	IndexOffset [▶ 2093]	Gets the index offset of the Symbol (Inherited from IProcessImageAddress [▶ 2092].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)

	Name	Description
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	IsVirtual [▶ 2094]	Gets a value indicating whether this instance is virtual. (Inherited from IProcessImageAddress [▶ 2092].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IAdsSymbol Interface [▶ 1379]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.7.1.1 IAdsSymbol.DataTypeId Property

DataType identifier of the Symbol AdsDataTypeId [▶ 574]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AdsDataTypeId DataTypeId { get; }
```

Property Value

Type: AdsDataTypeId [▶ 574]
Data type of the symbol.

Reference

IAdsSymbol Interface [▶ 1379]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.7.1.2 IAdsSymbol.ImageBaseAddress Property

Gets the AmsAddress [▶ 648] of the Process Image

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsAddress ImageBaseAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The address.

Reference

IAdsSymbol Interface [▶ 1379]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8 IAdsSymbolLoader Interface

Symbol Loader interface

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAdsSymbolLoader : ISymbolLoader,
    ISymbolProvider, ISymbolServer
```

The IAdsSymbolLoader type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types. (Inherited from ISymbolLoader [▶ 2200].)
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultNotificationSettings [▶ 1385]	Gets/Sets the default notification settings for this SymbolLoader
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	ImageBaseAddress [▶ 1386]	Gets the image base address.
	PlatformPointerSize [▶ 1386]	Gets the (byte) size of Pointers on the attached platform system.
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)
	Settings [▶ 2202]	Gets or sets the access Method (Inherited from ISymbolLoader [▶ 2200].)
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8.1 IAdsSymbolLoader Properties

The IAdsSymbolLoader [▶ 1383] type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types. (Inherited from ISymbolLoader [▶ 2200].)
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultNotificationSettings [▶ 1385]	Gets/Sets the default notification settings for this SymbolLoader

	Name	Description
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	ImageBaseAddress [▶ 1386]	Gets the image base address.
	PlatformPointerSize [▶ 1386]	Gets the (byte) size of Pointers on the attached platform system.
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)
	Settings [▶ 2202]	Gets or sets the access Method (Inherited from ISymbolLoader [▶ 2200].)
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Reference

IAdsSymbolLoader Interface [▶ 1383]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8.1.1 IAdsSymbolLoader.DefaultNotificationSettings Property

Gets/Sets the default notification settings for this SymbolLoader

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
INotificationSettings DefaultNotificationSettings { get; set; }
```

Property Value

Type: INotificationSettings [▶ 972]
The default notification settings.

Remarks

The Default notification Settings can be set on the SymbolLoader and is used as default on the different Symbols. On the symbol itself the NotificationSettings [▶ 979] can be overridden.

Examples

Setting the DefaultNotificationSettings on the IAdsSymbolLoader [▶ 1383] object:

Set DefaultNotificationSettings

```
// Create AdsClient object
using (AdsClient client = new AdsClient())
{
    // No automatic Synchronization (necessary for Console applications without message loop)
    //client.Synchronize = false;

    // Connect to client
    client.Connect(address);
```

```
    // Usage of 'dynamic' type/symbol loader
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree, ValueAcces
sMode.IndexGroupOffsetPreferred);
    IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

    // Set the DefaultNotification Properties
    dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.ClientOnChange, 20
0, 2000);

    // Determine the symbols
    dynamic dynamicSymbols = ((IDynamicSymbolLoader)dynLoader).SymbolsDynamic;

    // Task 1 Symbol (build in symbol)
    dynamic task1Symbol = dynamicSymbols._TaskInfo[1];

    // CycleCount Symbol
    dynamic cycleCountSymbol = task1Symbol.CycleCount;

    // Override Notification Setting for Cycle Count Symbol
    cycleCountSymbol.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 250, 0);

    // Register Dynamic Value Changed event.
    cycleCountSymbol.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChanged
);
```

Reference

IAdsSymbolLoader Interface [▶ 1383]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8.1.2 IAdsSymbolLoader.ImageBaseAddress Property

Gets the image base address.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
AmsAddress ImageBaseAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The image base address.

Reference

IAdsSymbolLoader Interface [▶ 1383]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8.1.3 IAdsSymbolLoader.PlatformPointerSize Property

Gets the (byte) size of Pointers on the attached platform system.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int PlatformPointerSize { get; }
```

Property Value

Type: Int32
The size of the platform pointer.

Reference

IAdsSymbolLoader Interface [▶ 1383]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.8.2 IAdsSymbolLoader Methods

The IAdsSymbolLoader [▶ 1383] type exposes the following members.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

IAdsSymbolLoader Interface [▶ 1383]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.9 IContextMaskProvider Interface

Interface IContextMaskProvider

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IContextMaskProvider
```

The IContextMaskProvider type exposes the following members.

Properties

	Name	Description
	ContextMask [▶ 1388]	Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications.

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.9.1 IContextMaskProvider Properties

The IContextMaskProvider [▶ 1387] type exposes the following members.

Properties

	Name	Description
	ContextMask [▶ 1388]	Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications.

Reference

IContextMaskProvider Interface [▶ 1387]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.9.1.1 IContextMaskProvider.ContextMask Property

Gets the ContextMask of the symbol, indicating the task the variable belongs to. If ContextMask is not zero use CyclicInContext [▶ 639] or OnChangeInContext [▶ 639] to add notifications.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
byte ContextMask { get; }
```

Property Value

Type: Byte

Reference

IContextMaskProvider Interface [▶ 1387]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10 Instance Class

Instance implementation

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.Instance
    TwinCAT.Ads.TypeSystem.Field [▶ 1374]
    TwinCAT.Ads.TypeSystem.Symbol [▶ 1469]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Instance : IInstance, IBitSize
```

The Instance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes.
	BitSize [▶ 1392]	Gets the size of this Instance in bits.
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance.
	Comment [▶ 1393]	Gets the comment.
	ContextMask [▶ 1394]	Gets the context mask of this instance.
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value.
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent.
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference.
	IsReadOnly [▶ 1398]	Indicates that this instance is read only.
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference.
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static.
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer.
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag.
	Namespace [▶ 1401]	Gets the namespace name.
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396]

	Name	Description
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Methods

	Name	Description
S	AlignTypeName [▶ 1403]	Aligns the type name
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance
	OnSetInstanceName [▶ 1404]	Sets a new InstanceName InstancePath
	SetAttributes [▶ 1404]	Sets the type attributes
	SetContextMask [▶ 1405]	Sets the context mask.
	ToString [▶ 1405]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1 Instance Properties

The Instance [▶ 1388] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes.
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits.
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance.
	Comment [▶ 1393]	Gets the comment.
	ContextMask [▶ 1394]	Gets the context mask of this instance.

	Name	Description
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value.
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent.
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference.
	IsReadOnly [▶ 1398]	Indicates that this instance is read only.
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference.
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static.
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer.
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag.
	Namespace [▶ 1401]	Gets the namespace name.
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396]
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.1 Instance.Attributes Property

Gets the Type Attributes.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection [▶ 2211]
The attributes.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.2 Instance.BitSize Property

Gets the size of this Instance [▶ 1388] in bits.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual int BitSize { get; }
```

Property Value

Type: Int32
The size of the bit.

Implements

IBitSize.BitSize [▶ 1984]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.3 Instance.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ByteSize { get; }
```

Property Value

Type: Int32
The size of the byte.

Implements

IBitSize.ByteSize [▶ 1984]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.4 Instance.Category Property

Gets the the DataTypeCategory [▶ 1649] of the Instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory Category { get; protected set; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The category.

Remarks

Corresponds to the Category [▶ 1988]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.5 Instance.Comment Property

Gets the comment.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Comment { get; }
```

Property Value

Type: String
The comment.

Implements

IInstance.Comment [▶ 2053]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.6 Instance.ContextMask Property

Gets the context mask of this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte ContextMask { get; }
```

Property Value

Type: Byte

Remarks

The Size of the internal data is 4-Bit

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.7 Instance.DataType Property

Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType DataType { get; protected set; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Implements

IInstance.DataType [▶ 2054]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.8 Instance.HasValue Property

Gets a value indicating whether this instance has a value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool HasValue { get; }
```

Property Value

Type: Boolean
true if this instance has value; otherwise, false.

Remarks

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.9 Instance.InstanceName Property

Gets the name of the instance (without periods (.)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string InstanceName { get; protected set; }
```

Property Value

Type: String
The name of the instance.

Implements

IInstance.InstanceName [▶ 2054]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.10 Instance.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.))

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Implements

IInstance.InstancePath [▶ 2055]

Remarks

If this path is relative or absolute depends on the context. IMember [▶ 2065] are using relative paths, ISymbol [▶ 2176]s are using absolute ones.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.11 Instance.IsBitType Property

Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping; otherwise, false.

Implements

IBitSize.IsBitType [▶ 1984]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.12 Instance.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Implements

IBitSize.IsByteAligned [▶ 1985]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.13 Instance.IsPersistent Property

Indicates that this instance is persistent.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPersistent { get; }
```

Property Value

Type: Boolean

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.14 Instance.IsPointer Property

Gets a value indicating whether this instance is reference.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPointer { get; }
```

Property Value

Type: Boolean
true if this instance is reference; otherwise, false.

Implements

IInstance.IsPointer [▶ 2055]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.15 Instance.IsReadOnly Property

Indicates that this instance is read only.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean

Remarks

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied [▶ 575] error.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.16 Instance.IsReference Property

Gets a value indicating whether this instance is reference.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReference { get; }
```

Property Value

Type: Boolean
true if this instance is reference; otherwise, false.

Implements

IInstance.IsReference [▶ 2056]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.17 Instance.IsStatic Property

Gets a value indicating whether this IInstance [▶ 2052] is static.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsStatic { get; protected set; }
```

Property Value

Type: Boolean
true if this instance is static; otherwise, false.

Implements

IInstance.IsStatic [▶ 2056]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.18 Instance.IsTcComInterfacePointer Property

Indicates that this instance is a TcComInterfacePointer.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsTcComInterfacePointer { get; }
```

Property Value

Type: Boolean

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.19 Instance.IsTypeGuid Property

Indicates that this instance has set TypeGuid flag.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsTypeGuid { get; }
```

Property Value

Type: Boolean

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.20 Instance.Namespace Property

Gets the namespace name.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Namespace { get; }
```

Property Value

Type: String
The namespace.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.21 Instance.Size Property

Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Size { get; protected set; }
```

Property Value

Type: Int32
The size of the bit.

Implements

IBitSize.Size [▶ 1985]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.1.22 Instance.TypeName Property

Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string TypeName { get; protected set; }
```

Property Value

Type: String
The name of the type.

Implements

IInstance.TypeName [▶ 2056]

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2 Instance Methods

The Instance [▶ 1388] type exposes the following members.

Methods

	Name	Description
	AlignTypeName [▶ 1403]	Aligns the type name
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance [▶ 1388]
	OnSetInstanceName [▶ 1404]	Sets a new InstanceName InstancePath
	SetAttributes [▶ 1404]	Sets the type attributes

	Name	Description
	SetContextMask [▶ 1405]	Sets the context mask.
	ToString [▶ 1405]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.1 Instance.AlignTypeName Method

Aligns the type name

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected static string AlignTypeName(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the type.

Return Value

Type: String
System.String.

Exceptions

Exception	Condition
ArgumentException	Type name not valid!

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.2 Instance.OnGetSize Method

Handler function getting the size of the Instance [▶ 1388]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual int OnGetSize()
```

Return Value

Type: Int32
System.Int32.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.3 Instance.OnSetInstanceName Method

Sets a new InstanceName InstancePath

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnSetInstanceName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Instance name.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.4 Instance.SetAttributes Method

Sets the type attributes

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetAttributes(
    TypeAttributeCollection coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.TypeAttributeCollection [▶ 2418] The attributes.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.5 Instance.SetContextMask Method

Sets the context mask.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void SetContextMask(
    byte contextMask
)
```

Parameters

contextMask	Type: System.Byte The context mask.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	contextMask

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.10.2.6 Instance.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

Instance Class [▶ 1388]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11 Member Class

Represents a member of an StructType [▶ 1457]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.Instance [▶ 1388]
    TwinCAT.Ads.TypeSystem.Field [▶ 1374]
      TwinCAT.Ads.TypeSystem.Member
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class Member : Field,
    IMember, IField, IAttributedInstance, IInstance, IBitSize
```

The Member type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitOffset [▶ 1409]	Gets the bit offset.
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteOffset [▶ 1409]	Gets the byte offset.
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)

	Name	Description
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance [▶ 1388].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)
	Offset [▶ 1410]	Gets the offset of the Member within the parent StructType [▶ 1457] in bits or bytes dependent on IsBitType [▶ 1396]
	ParentType [▶ 1378]	Gets the Parent of this IField [▶ 2040]. (Inherited from Field [▶ 1374].)
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1378]	Gets the value encoding of this Field [▶ 1374] (Inherited from Field [▶ 1374].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1405]	Returns a String that represents this instance. (Inherited from Instance [▶ 1388].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11.1 Member Properties

The Member [▶ 1406] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitOffset [▶ 1409]	Gets the bit offset.
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteOffset [▶ 1409]	Gets the byte offset.
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1396]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from Instance [▶ 1388].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)

	Name	Description
	Offset [▶ 1410]	Gets the offset of the Member [▶ 1406] within the parent StructType [▶ 1457] in bits or bytes dependent on IsBitType [▶ 1396]
	ParentType [▶ 1378]	Gets the Parent of this IField [▶ 2040]. (Inherited from Field [▶ 1374].)
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1378]	Gets the value encoding of this Field [▶ 1374] (Inherited from Field [▶ 1374].)

Reference

Member Class [▶ 1406]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11.1.1 Member.BitOffset Property

Gets the bit offset.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int BitOffset { get; }
```

Property Value

Type: Int32
The bit offset.

Implements

IMember.BitOffset [▶ 2067]

Reference

Member Class [▶ 1406]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11.1.2 Member.ByteOffset Property

Gets the byte offset.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ByteOffset { get; }
```

Property Value

Type: Int32
The byte offset.

Implements

IMember.ByteOffset [▶ 2067]

Reference

Member Class [▶ 1406]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11.1.3 Member.Offset Property

Gets the offset of the Member [▶ 1406] within the parent StructType [▶ 1457] in bits or bytes dependent on IsBitType [▶ 1396]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Offset { get; }
```

Property Value

Type: Int32
The offset.

Implements

IMember.Offset [▶ 2068]

Reference

Member Class [▶ 1406]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.11.2 Member Methods

The Member [▶ 1406] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1405]	Returns a String that represents this instance. (Inherited from Instance [▶ 1388].)

Reference

Member Class [▶ 1406]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.12 PCCHType Class

Class PCCHType. This class cannot be inherited. Implements the PointerType [▶ 1414]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.PointerType [▶ 1414]
      TwinCAT.Ads.TypeSystem.PCCHType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class PCCHType : PointerType
```

The PCCHType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)

	Name	Description
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Inherited from PointerType [▶ 1414].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type. (Inherited from PointerType [▶ 1414].)
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype (Inherited from PointerType [▶ 1414].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.Ads.TypeSystem.PointerType [▶ 1414]

6.7.12.1 PCCHType Properties

The PCCHType [▶ 1411] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Inherited from PointerType [▶ 1414].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type. (Inherited from PointerType [▶ 1414].)
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype (Inherited from PointerType [▶ 1414].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

PCCHType Class [▶ 1411]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.12.2 PCCHType Methods

The PCCHType [▶ 1411] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

PCCHType Class [▶ 1411]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13 PointerType Class

Represents a pointer type.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.PointerType
      TwinCAT.Ads.TypeSystem.PCCHType [▶ 1411]
      TwinCAT.Ads.TypeSystem.PVoidType [▶ 1422]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class PointerType : DataType,
    IPointerType, IDataType, IBitSize
```

The PointerType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)

	Name	Description
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type.
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13.1 PointerType Properties

The PointerType [▶ 1414] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type.
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

PointerType Class [▶ 1414]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13.1.1 PointerType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override Type ManagedType { get; }
```

Property Value

Type: Type
Dot net type.

Reference

PointerType Class [▶ 1414]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13.1.2 PointerType.ReferencedType Property

Gets the the referenced type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the referenced.

Implements

IPointerType.ReferencedType [▶ 2088]

Reference

PointerType Class [▶ 1414]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13.1.3 PointerType.ReferenceTypeName Property

Gets the name of the referenced datatype

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReferenceTypeName { get; }
```

Property Value

Type: String
The name of the reference dattype.

Implements

IPointerType.ReferenceTypeName [▶ 2089]

Reference

PointerType Class [▶ 1414]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.13.2 PointerType Methods

The PointerType [▶ 1414] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

PointerType Class [▶ 1414]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.14 PrimitiveType Class

Class PrimitiveType.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.PrimitiveType
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class PrimitiveType : DataType,
    IPrimitiveType, IDataType, IBitSize
```

The PrimitiveType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)

	Name	Description
	PrimitiveFlags [▶ 1421]	Indicates types of different PrimitiveTypes with flags.
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.Ads.TypeSystem.DataType [▶ 1349]

6.7.14.1 PrimitiveType Properties

The PrimitiveType [▶ 1418] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)

	Name	Description
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	PrimitiveFlags [▶ 1421]	Indicates types of different PrimitiveTypes with flags.
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

PrimitiveType Class [▶ 1418]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.14.1.1 PrimitiveType.PrimitiveFlags Property

Indicates types of different PrimitiveTypes with flags.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public PrimitiveTypeFlags PrimitiveFlags { get; }
```

Property Value

Type: PrimitiveTypeFlags [▶ 2289]
The primitive flags.

Implements

IPrimitiveType.PrimitiveFlags [▶ 2091]

Reference

PrimitiveType Class [▶ 1418]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.14.2 PrimitiveType Methods

The PrimitiveType [▶ 1418] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

PrimitiveType Class [▶ 1418]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.15 PVoidType Class

Class PVoidType. This class cannot be inherited. Implements the PointerType [▶ 1414]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.PointerType [▶ 1414]
      TwinCAT.Ads.TypeSystem.PVoidType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class PVoidType : PointerType
```

The PVoidType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)

	Name	Description
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Inherited from PointerType [▶ 1414].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type. (Inherited from PointerType [▶ 1414].)
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype (Inherited from PointerType [▶ 1414].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.Ads.TypeSystem.PointerType [▶ 1414]

6.7.15.1 PVoidType Properties

The PVoidType [▶ 1422] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1417]	Gets the corresponding .NET Type if attached. (Inherited from PointerType [▶ 1414].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1417]	Gets the the referenced type. (Inherited from PointerType [▶ 1414].)
	ReferenceTypeName [▶ 1417]	Gets the name of the referenced datatype (Inherited from PointerType [▶ 1414].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

PVoidType Class [▶ 1422]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.15.2 PVoidType Methods

The PVoidType [▶ 1422] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

PVoidType Class [▶ 1422]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16 ReferenceType Class

Represents a reference type

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.ReferenceType
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class ReferenceType : DataType,
    IReferenceType, IDataType, IBitSize
```

The ReferenceType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)

	Name	Description
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1428]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Overrides DataType.IsContainer [▶ 1357].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1429]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1429]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1430]	Gets the the referenced type.
	ReferencedTypeName [▶ 1430]	Gets the name of the referenced type.
	ResolvedByteSize [▶ 1431]	Gets the size of the resolved byte.
	ResolvedCategory [▶ 1431]	Gets the resolved category.
	ResolvedType [▶ 1432]	Gets the type of the resolved.
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1 ReferenceType Properties

The ReferenceType [▶ 1425] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1428]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Overrides DataType.IsContainer [▶ 1357].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1429]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Overrides DataType.IsPrimitive [▶ 1358].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1429]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	ReferencedType [▶ 1430]	Gets the the referenced type.

	Name	Description
	ReferencedTypeName [▶ 1430]	Gets the name of the referenced type.
	ResolvedByteSize [▶ 1431]	Gets the size of the resolved byte.
	ResolvedCategory [▶ 1431]	Gets the resolved category.
	ResolvedType [▶ 1432]	Gets the type of the resolved.
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.1 ReferenceType.IsContainer Property

Gets a value indicating whether this IDataType [▶ 1986] is a container type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool IsContainer { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

IDataType.IsContainer [▶ 1990]
IDataType.IsContainer [▶ 1990]

Remarks

Container Types are all types that contain SubElements like

- Array [▶ 1649]
- Pointer [▶ 1649]
- Union [▶ 1649]
- Struct [▶ 1649]
- Function [▶ 1649]
- FunctionBlock [▶ 1649]
- Program [▶ 1649]

and the Alias [▶ 1649] and Reference [▶ 1649] types, if they have a container type as base type.

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.16.1.2 ReferenceType.IsPrimitive Property

Gets a value indicating whether this IDataType [▶ 1986] is primitive

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IDataType.IsPrimitive [▶ 1991]
IDataType.IsPrimitive [▶ 1991]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.3 ReferenceType.ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override Type ManagedType { get; }
```

Property Value

Type: Type
Dot net type.

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.4 ReferenceType.ReferencedType Property

Gets the the referenced type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the referenced.

Implements

IReferenceType.ReferencedType [▶ 2102]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.5 ReferenceType.ReferencedTypeName Property

Gets the name of the referenced type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReferencedTypeName { get; }
```

Property Value

Type: String
The name of the referenced type.

Implements

IReferenceType.ReferencedTypeName [▶ 2102]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.6 ReferenceType.ResolvedByteSize Property

Gets the size of the resolved byte.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ResolvedByteSize { get; }
```

Property Value

Type: Int32
The size of the resolved byte.

Implements

IReferenceType.ResolvedByteSize [▶ 2102]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.7 ReferenceType.ResolvedCategory Property

Gets the resolved category.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory ResolvedCategory { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The resolved category.

Implements

IReferenceType.ResolvedCategory [▶ 2103]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.1.8 ReferenceType.ResolvedType Property

Gets the type of the resolved.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ResolvedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the resolved.

Implements

IReferenceType.ResolvedType [▶ 2103]

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.16.2 ReferenceType Methods

The ReferenceType [▶ 1425] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

ReferenceType Class [▶ 1425]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17 RpcMethod Class

RPC Method Description

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.RpcMethod
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RpcMethod : IRpcMethod
```

The RpcMethod type exposes the following members.

Properties

	Name	Description
	Comment [▶ 1434]	Gets the Method comment.
	InParameters [▶ 1435]	Gets the In-Parameters of the IRpcMethod [▶ 2123]
	IsVoid [▶ 1435]	Gets a value indicating whether this IRpcMethod [▶ 2123] has no return parameter
	Name [▶ 1436]	Gets the name of the method
	OutParameters [▶ 1436]	Gets the Out-Parameters of the IRpcMethod [▶ 2123]
	Parameters [▶ 1437]	Gets all parameters (In, Out and ref parameters) of the . [▶ 2123]
	ReturnAlignSize [▶ 1437]	Gets the size of the biggest element in bytes for Alignment
	ReturnType [▶ 1438]	Gets the return type.
	ReturnTypeSize [▶ 1438]	Gets the Byte size of the return type.
	VTableIndex [▶ 1439]	Gets the V-table index of the method.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	ToString [▶ 1440]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1 RpcMethod Properties

The RpcMethod [▶ 1433] type exposes the following members.

Properties

	Name	Description
	Comment [▶ 1434]	Gets the Method comment.
	InParameters [▶ 1435]	Gets the In-Parameters of the IRpcMethod [▶ 2123]
	IsVoid [▶ 1435]	Gets a value indicating whether this IRpcMethod [▶ 2123] has no return parameter
	Name [▶ 1436]	Gets the name of the method
	OutParameters [▶ 1436]	Gets the Out-Parameters of the IRpcMethod [▶ 2123]
	Parameters [▶ 1437]	Gets all parameters (In, Out and ref parameters) of the . [▶ 2123]
	ReturnAlignSize [▶ 1437]	Gets the size of the biggest element in bytes for Alignment
	ReturnType [▶ 1438]	Gets the return type.
	ReturnTypeSize [▶ 1438]	Gets the Byte size of the return type.
	VTableIndex [▶ 1439]	Gets the V-table index of the method.

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.1 RpcMethod.Comment Property

Gets the Method comment.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Comment { get; }
```

Property Value

Type: String
The comment.

Implements

IRpcMethod.Comment [▶ 2124]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.2 RpcMethod.InParameters Property

Gets the In-Parameters of the IRpcMethod [▶ 2123]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameterCollection InParameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The In- and Ref-Parameters

Implements

IRpcMethod.InParameters [▶ 2124]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.3 RpcMethod.IsVoid Property

Gets a value indicating whether this IRpcMethod [▶ 2123] has no return parameter

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsVoid { get; }
```

Property Value

Type: Boolean
true if this instance is void; otherwise, false.

Implements

IRpcMethod.IsVoid [▶ 2125]

Exceptions

Exception	Condition
NotImplementedException	

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.4 RpcMethod.Name Property

Gets the name of the method

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Implements

IRpcMethod.Name [▶ 2125]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.5 RpcMethod.OutParameters Property

Gets the Out-Parameters of the IRpcMethod [▶ 2123]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameterCollection OutParameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The In- and Ref-Parameters

Implements

IRpcMethod.OutParameters [▶ 2126]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.6 RpcMethod.Parameters Property

Gets all parameters (In, Out and ref parameters) of the . [▶ 2123]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameterCollection Parameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The parameters.

Implements

IRpcMethod.Parameters [▶ 2126]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.7 RpcMethod.ReturnAlignSize Property

Gets the size of the biggest element in bytes for Alignment

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReturnAlignSize { get; }
```

Property Value

Type: Int32
The size of the return align.

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.8 RpcMethod.ReturnType Property

Gets the return type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string ReturnType { get; }
```

Property Value

Type: String
Return type.

Implements

IRpcMethod.ReturnType [▶ 2126]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.9 RpcMethod.ReturnTypeSize Property

Gets the Byte size of the return type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ReturnTypeSize { get; }
```

Property Value

Type: Int32
The size of the return type.

Implements

IRpcMethod.ReturnTypeSize [▶ 2127]

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.1.10 RpcMethod.VTableIndex Property

Gets the V-table index of the method.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int VTableIndex { get; }
```

Property Value

Type: Int32
The index of the v table.

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.2 RpcMethod Methods

The RpcMethod [▶ 1433] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1440]	Returns a String that represents this instance. (Overrides Object.ToString..)

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.17.2.1 RpcMethod.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

RpcMethod Class [▶ 1433]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18 RpcMethodParameter Class

Class RpcMethodParameter.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.RpcMethodParameter
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RpcMethodParameter : IRpcMethodParameter
```

The RpcMethodParameter type exposes the following members.

Properties

	Name	Description
	AlignSize [▶ 1442]	Gets the size of biggest element for alignment
	Comment [▶ 1443]	Gets the Parameter Comment.
	HasLengthIsParameter [▶ 1443]	Gets a value indicating whether this instance has a related LengthIs Parameter.
	LengthIsParameterIndex [▶ 1443]	Gets the index of the LengthIs parameter (within the MethodParameter List)
	Name [▶ 1444]	Gets the Parameter Name
	ParameterFlags [▶ 1445]	Gets the parameter flags.
	Size [▶ 1445]	Gets the size of the RpcMethodParameter
	TypeGuid [▶ 1446]	Gets the Unique identifier of the parameters data type.
	TypeName [▶ 1446]	Gets the Data type of the Parameter

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1 RpcMethodParameter Properties

The RpcMethodParameter [▶ 1440] type exposes the following members.

Properties

	Name	Description
	AlignSize [▶ 1442]	Gets the size of biggest element for alignment
	Comment [▶ 1443]	Gets the Parameter Comment.
	HasLengthIsParameter [▶ 1443]	Gets a value indicating whether this instance has a related LengthIs Parameter.
	LengthIsParameterIndex [▶ 1443]	Gets the index of the LengthIs parameter (within the MethodParameter List)
	Name [▶ 1444]	Gets the Parameter Name
	ParameterFlags [▶ 1445]	Gets the parameter flags.
	Size [▶ 1445]	Gets the size of the RpcMethodParameter [▶ 1440]
	TypeGuid [▶ 1446]	Gets the Unique identifier of the parameters data type.
	TypeName [▶ 1446]	Gets the Data type of the Parameter

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.1 RpcMethodParameter.AlignSize Property

Gets the size of biggest element for alignment

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int AlignSize { get; }
```

Property Value

Type: Int32
The size of the align.

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.2 RpcMethodParameter.Comment Property

Gets the Parameter Comment.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Comment { get; }
```

Property Value

Type: String
The comment.

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.3 RpcMethodParameter.HasLengthIsParameter Property

Gets a value indicating whether this instance has a related LengthIs Parameter.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool HasLengthIsParameter { get; }
```

Property Value

Type: Boolean
true if this instance has a LengthIs parameter; otherwise, false.

Implements

IRpcMethodParameter.HasLengthIsParameter [▶ 2135]

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.4 RpcMethodParameter.LengthIsParameterIndex Property

Gets the index of the LengthIs parameter (within the MethodParameter List)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int LengthIsParameterIndex { get; }
```

Property Value

Type: Int32
The index of the length is parameter.

Implements

IRpcMethodParameter.LengthIsParameterIndex [▶ 2135]

Remarks

This field references to the Parameter that defines the length for this generic one. Equally to the marshalling attributes of COM (sizeof, length) this enables to transport parameter of type (PVOID)

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.5 RpcMethodParameter.Name Property

Gets the Parameter Name

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Implements

IRpcMethodParameter.Name [▶ 2135]

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.6 RpcMethodParameter.ParameterFlags Property

Gets the parameter flags.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MethodParamFlags ParameterFlags { get; }
```

Property Value

Type: MethodParamFlags [▶ 2288]
The parameter flags.

Implements

IRpcMethodParameter.ParameterFlags [▶ 2136]

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.7 RpcMethodParameter.Size Property

Gets the size of the RpcMethodParameter [▶ 1440]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Size { get; }
```

Property Value

Type: Int32
The size.

Implements

IRpcMethodParameter.Size [▶ 2136]

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.8 RpcMethodParameter.TypeGuid Property

Gets the Unique identifier of the parameters data type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Guid TypeGuid { get; }
```

Property Value

Type: Guid
The type unique identifier.

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.1.9 RpcMethodParameter.TypeName Property

Gets the Data type of the Parameter

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string TypeName { get; }
```

Property Value

Type: String
The type.

Implements

IRpcMethodParameter.TypeName [▶ 2137]

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.18.2 RpcMethodParameter Methods

The RpcMethodParameter [▶ 1440] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

RpcMethodParameter Class [▶ 1440]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.19 RpcStructType Class

StructType which is callable by RPC Methods.

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.StructType [▶ 1457]
      TwinCAT.Ads.TypeSystem.RpcStructType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class RpcStructType : StructType,
    IRpcCallableType
```

The RpcStructType type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 1460]	Gets all members (down the derivation hierarchy) (Inherited from StructType [▶ 1457].)
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1460]	Gets the structs Base Type (Null if not derived). (Inherited from StructType [▶ 1457].)
	BaseTypeName [▶ 1461]	Gets the the Name of the Base class (if derived) (Inherited from StructType [▶ 1457].)

	Name	Description
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	HasRpcMethods [▶ 1450]	Gets a value indicating whether this instance has RPC Methods. (Overrides StructType.HasRpcMethods [▶ 1461].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsDerived [▶ 1462]	Gets a value indicating whether this instance is derived. (Inherited from StructType [▶ 1457].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Members [▶ 1462]	Gets a read only collection of the Members [▶ 2065] of the IStructType [▶ 2162]. (Inherited from StructType [▶ 1457].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	RpcMethods [▶ 1451]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.19.1 RpcStructType Properties

The RpcStructType [▶ 1447] type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 1460]	Gets all members (down the derivation hierarchy) (Inherited from StructType [▶ 1457].)
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1460]	Gets the structs Base Type (Null if not derived). (Inherited from StructType [▶ 1457].)
	BaseTypeName [▶ 1461]	Gets the the Name of the Base class (if derived) (Inherited from StructType [▶ 1457].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	HasRpcMethods [▶ 1450]	Gets a value indicating whether this instance has RPC Methods. (Overrides StructType.HasRpcMethods [▶ 1461].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsDerived [▶ 1462]	Gets a value indicating whether this instance is derived. (Inherited from StructType [▶ 1457].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)

	Name	Description
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Members [▶ 1462]	Gets a read only collection of the Members [▶ 2065] of the IStructType [▶ 2162]. (Inherited from StructType [▶ 1457].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	RpcMethods [▶ 1451]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

RpcStructType Class [▶ 1447]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.19.1.1 RpcStructType.HasRpcMethods Property

Gets a value indicating whether this instance has RPC Methods.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this instance has methods; otherwise, false.

Implements

IStructType.HasRpcMethods [▶ 2166]

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

RpcStructType Class [▶ 1447]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.19.1.2 RpcStructType.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodCollection RpcMethods { get; }
```

Property Value

Type: IRpcMethodCollection [▶ 2127]
The methods.

Implements

IRpcCallableType.RpcMethods [▶ 2122]

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

RpcStructType Class [▶ 1447]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.19.2 RpcStructType Methods

The RpcStructType [▶ 1447] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

RpcStructType Class [▶ 1447]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20 StringType Class

String DataType

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.StringType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class StringType : DataType,
    IStringType, IDataType, IBitSize
```

The StringType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Encoding [▶ 1454]	Gets the encoding of the String (Encoding.Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsFixedLength [▶ 1455]	Gets a value indicating whether the string is of fixed length.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)

	Name	Description
	Length [▶ 1455]	Gets the number of characters within the string.
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1456]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.1 StringType Properties

The StringType [▶ 1452] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Encoding [▶ 1454]	Gets the encoding of the String (Encoding.Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)

	Name	Description
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsFixedLength [▶ 1455]	Gets a value indicating whether the string is of fixed length.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	Length [▶ 1455]	Gets the number of characters within the string.
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.1.1 StringType.Encoding Property

Gets the encoding of the String (Encoding.Default (Ansi Codepage, STRING) or Encoding.UNICODE (WSTRING))

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding Encoding { get; }
```

Property Value

Type: Encoding
The encoding.

Implements

IStringType.Encoding [▶ 2157]

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.1.2 StringType.IsFixedLength Property

Gets a value indicating whether the string is of fixed length.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsFixedLength { get; }
```

Property Value

Type: Boolean
true if this instance is fixed length; otherwise, false.

Implements

IStringType.IsFixedLength [▶ 2157]

Exceptions

Exception	Condition
NotImplementedException	

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.1.3 StringType.Length Property

Gets the number of characters within the string.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Length { get; }
```

Property Value

Type: Int32
The length.

Implements

IStringType.Length [▶ 2158]

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.2 StringType Methods

The StringType [▶ 1452] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1456]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.20.2.1 StringType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

StringType Class [▶ 1452]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21 StructType Class

Represents a struct type

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.StructType
      TwinCAT.Ads.TypeSystem.RpcStructType [▶ 1447]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class StructType : DataType, IStructType,
    IDataType, IBitSize
```

The StructType type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 1460]	Gets all members (down the derivation hierarchy)
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1460]	Gets the structs Base Type (Null if not derived).
	BaseTypeName [▶ 1461]	Gets the the Name of the Base class (if derived)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	HasRpcMethods [▶ 1461]	Gets a value indicating whether this StructType has RPC Methods.
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)

	Name	Description
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsDerived [▶ 1462]	Gets a value indicating whether this instance is derived.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Members [▶ 1462]	Gets a read only collection of the Members [▶ 2065] of the IStructType [▶ 2162].
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1 StructType Properties

The StructType [▶ 1457] type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 1460]	Gets all members (down the derivation hierarchy)
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1460]	Gets the structs Base Type (Null if not derived).
	BaseTypeName [▶ 1461]	Gets the the Name of the Base class (if derived)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	HasRpcMethods [▶ 1461]	Gets a value indicating whether this StructType [▶ 1457] has RPC Methods.
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsDerived [▶ 1462]	Gets a value indicating whether this instance is derived.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Members [▶ 1462]	Gets a read only collection of the Members [▶ 2065] of the IStructType [▶ 2162].
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.1 StructType.AllMembers Property

Gets all members (down the derivation hierarchy)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IMemberCollection AllMembers { get; }
```

Property Value

Type: IMemberCollection [▶ 2068]
All members.

Implements

IStructType.AllMembers [▶ 2165]

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.2 StructType.BaseType Property

Gets the structs Base Type (Null if not derived).

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]

Implements

IStructType.BaseType [▶ 2165]

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.3 StructType.BaseTypeName Property

Gets the the Name of the Base class (if derived)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string BaseTypeName { get; }
```

Property Value

Type: String
Empty if not derived.

Implements

IStructType.BaseTypeName [▶ 2165]

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.4 StructType.HasRpcMethods Property

Gets a value indicating whether this StructType [▶ 1457] has RPC Methods.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this type has methods; otherwise, false.

Implements

IStructType.HasRpcMethods [▶ 2166]

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.5 StructType.IsDerived Property

Gets a value indicating whether this instance is derived.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsDerived { get; }
```

Property Value

Type: Boolean
true if this instance is derived; otherwise, false.

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.1.6 StructType.Members Property

Gets a read only collection of the Members [▶ 2065] of the IStructType [▶ 2162].

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IMemberCollection Members { get; }
```

Property Value

Type: IMemberCollection [▶ 2068]
The members as read only collection.

Implements

IStructType.Members [▶ 2166]

Remarks

If the IStructType [▶ 2162] is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers [▶ 2165] property.

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.21.2 StructType Methods

The StructType [▶ 1457] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

StructType Class [▶ 1457]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22 SubRangeType.T. Class

Represents a SubRangType

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.SubRangeType.T.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class SubRangeType<T> : DataType,
    ISubRangeType<T>, ISubRangeType, IDataType, IBitSize
where T : struct, new()
```

Type Parameters

T

The SubRangeType.T. type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1466]	Gets the the base type.
	BaseTypeName [▶ 1466]	Gets the name of the base type.
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	LowerBound [▶ 1467]	Gets the lower bound.
	ManagedType [▶ 1467]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)
	UpperBound [▶ 1468]	Gets the upper bound.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1 SubRangeType.T. Properties

The SubRangeType.T. [▶ 1463] generic type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BaseType [▶ 1466]	Gets the the base type.
	BaseTypeName [▶ 1466]	Gets the name of the base type.
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)

	Name	Description
	LowerBound [▶ 1467]	Gets the lower bound.
	ManagedType [▶ 1467]	Gets the corresponding .NET Type if attached. (Overrides DataType.ManagedType [▶ 1359].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)
	UpperBound [▶ 1468]	Gets the upper bound.

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1.1 SubRangeType.T..BaseType Property

Gets the the base type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the referenced.

Implements

ISubRangeType.BaseType [▶ 2172]

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1.2 SubRangeType.T..BaseTypeName Property

Gets the name of the base type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string BaseTypeName { get; }
```

Property Value

Type: String
The name of the base type.

Implements

ISubRangeType.BaseTypeName [▶ 2173]

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1.3 SubRangeType.T..LowerBound Property

Gets the lower bound.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T LowerBound { get; }
```

Property Value

Type: T [▶ 1463]
The lower bound.

Implements

ISubRangeType.T..LowerBound [▶ 2176]

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1.4 SubRangeType.T..ManagedType Property

Gets the corresponding .NET Type if attached.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override Type ManagedType { get; }
```

Property Value

Type: Type
Dot net type.

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.1.5 SubRangeType.T..UpperBound Property

Gets the upper bound.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T UpperBound { get; }
```

Property Value

Type: T [▶ 1463]
The upper bound.

Implements

ISubRangeType.T..UpperBound [▶ 2176]

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.22.2 SubRangeType.T. Methods

The SubRangeType.T. [▶ 1463] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

SubRangeType.T. Class [▶ 1463]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23 Symbol Class

Symbol class

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.Instance [▶ 1388]
    TwinCAT.Ads.TypeSystem.Symbol
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Symbol : Instance, IValueSymbol,
    IValueRawSymbol, IHierarchicalSymbol, ISymbol, IAttributedInstance, IInstance,
    IBitSize, IValueAnySymbol, IValueAccessorProvider, ISymbolFactoryServicesProvider, IAdsSymbol,
    IProcessImageAddress, IContextMaskProvider
```

The Symbol type exposes the following members.

Properties

	Name	Description
	AccessRights [▶ 1476]	Gets the access rights.
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	Connection [▶ 1477]	Gets the connection that produces values for this IValueSymbol [▶ 2254]
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)

	Name	Description
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)
	ImageBaseAddress [▶ 1477]	Gets the AmsAddress [▶ 648] of the Process Image
	IndexGroup [▶ 1478]	Gets the index group of the Symbol
	IndexOffset [▶ 1478]	Gets the index offset of the Symbol
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1479]	Gets the relative / absolute access path to the instance (with periods (.)) (Overrides Instance.InstancePath [▶ 1396].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsContainerType [▶ 1479]	Gets a value indicating whether the Symbols datatype is a Container type.
	IsDereferencedPointer [▶ 1480]	Gets or sets a value indicating whether an ancestor is a dereferenced Pointer
	IsDereferencedReference [▶ 1481]	Gets or sets a value indicating whether an ancestor is a dereferenced Reference
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsPrimitiveType [▶ 1481]	Gets a value indicating whether this instance is primitive.
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsRecursive [▶ 1482]	Gets a value indicating whether this instance is recursive.
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	IsVirtual [▶ 1482]	Gets a value indicating whether this instance is virtual.
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)
	NotificationSettings [▶ 1483]	Gets or sets the notification settings.
	Parent [▶ 1483]	Gets the parent Symbol

	Name	Description
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	SubSymbolCount [▶ 1484]	Gets the number of SubSymbols
	SubSymbols [▶ 1484]	Gets the SubSymbols of the ISymbol [▶ 2176]
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1485]	Gets the value encoding.

Methods

	Name	Description
	EnsureRights [▶ 1488]	Ensures that the AccessRights are matched.
	Equals [▶ 1489]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1489]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance [▶ 1388] (Inherited from Instance [▶ 1388].)
	OnReadRawValue [▶ 1490]	Handler function for reading the raw value
	OnReadRawValueAsync [▶ 1490]	Handler function for reading the raw value
	OnReadValue [▶ 1491]	Handler function for reading the dynamic value.
	OnReadValueAsync [▶ 1492]	Handler function for reading the dynamic value.
	OnSetInstanceName [▶ 1492]	Sets a new InstanceName InstancePath (Overrides Instance.OnSetInstanceName(String) [▶ 1404].)
	OnTryReadValue [▶ 1493]	Handler function for reading the dynamic value.
	OnTryWriteValue [▶ 1493]	Handler function for writing the dynamic value
	OnWriteRawValue [▶ 1494]	
	OnWriteRawValueAsync [▶ 1494]	
	OnWriteValue [▶ 1495]	Handler function for writing the dynamic value

	Name	Description
	OnWriteValueAsync [▶ 1496]	Handler function for writing the dynamic value
	ReadAnyValue(Type) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadAnyValue(Type, Int32) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadAnyValueAsync [▶ 1498]	Reads the (AnyType) value asynchronously.
	ReadRawValue. [▶ 1499]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1500]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValueAsync [▶ 1501]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.
	ReadValue. [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValue(Int32) [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValueAsync [▶ 1503]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously.
	SetAttributes [▶ 1404]	Sets the type attributes (Inherited from Instance [▶ 1388].)
	SetContextMask [▶ 1405]	Sets the context mask. (Inherited from Instance [▶ 1388].)
	SetParent [▶ 1504]	Sets the parent symbol.
	ToString [▶ 1505]	Returns a String that represents this instance. (Overrides Instance.ToString. [▶ 1405].)
	TryReadValue [▶ 1505]	Reads the Value of the IValueSymbol [▶ 2254]
	TryWriteValue [▶ 1506]	Writes the Value of the IValueSymbol [▶ 2254]
	UpdateAnyValue(Object.) [▶ 1507]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 1508]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteRawValue(.Byte.) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 1510]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

	Name	Description
	WriteValue(Object) [▶ 1511]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValue(Object, Int32) [▶ 1512]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValueAsync [▶ 1513]	Writes the Value of the IValueSymbol [▶ 2254]

Events

	Name	Description
	RawValueChanged [▶ 1514]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.
	ValueChanged [▶ 1514]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Operators

	Name	Description
	Equality [▶ 1515]	Operator==
	Inequality [▶ 1516]	Implements the != operator.

Fields

	Name	Description
	syncObject [▶ 1517]	Synchronization object

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

Remarks

A Symbol is a (named) memory object within the Process Image with a fixed address indicated by Index Group and Index Offset. Symbols can optionally be addressed by instance path and are bound to a specific DataType [▶ 1349].

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.Ads.TypeSystem.Instance [▶ 1388]

TwinCAT.TypeSystem.IValueSymbol [▶ 2254]

TwinCAT.TypeSystem.IValueAnySymbol [▶ 2235]

TwinCAT.TypeSystem.IValueAccessorProvider [▶ 2234]

TwinCAT.TypeSystem.ISymbolFactoryServicesProvider [▶ 2198]

TwinCAT.TypeSystem.IHierarchicalSymbol [▶ 2048]

ISymbolValueChangeNotify

TwinCAT.Ads.TypeSystem.IContextMaskProvider [▶ 1387]

IInstanceInternal

ISymbolInternal

TwinCAT.Ads.TypeSystem.IAdsSymbol [▶ 1379]

6.7.23.1 Symbol Properties

The Symbol [▶ 1469] type exposes the following members.

Properties

	Name	Description
	AccessRights [▶ 1476]	Gets the access rights.

	Name	Description
	Attributes [▶ 1391]	Gets the Type Attributes. (Inherited from Instance [▶ 1388].)
	BitSize [▶ 1392]	Gets the size of this Instance [▶ 1388] in bits. (Inherited from Instance [▶ 1388].)
	ByteSize [▶ 1392]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from Instance [▶ 1388].)
	Category [▶ 1393]	Gets the the DataTypeCategory [▶ 1649] of the Instance. (Inherited from Instance [▶ 1388].)
	Comment [▶ 1393]	Gets the comment. (Inherited from Instance [▶ 1388].)
	Connection [▶ 1477]	Gets the connection that produces values for this IValueSymbol [▶ 2254]
	ContextMask [▶ 1394]	Gets the context mask of this instance. (Inherited from Instance [▶ 1388].)
	DataType [▶ 1394]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	HasValue [▶ 1395]	Gets a value indicating whether this instance has a value. (Inherited from Instance [▶ 1388].)
	ImageBaseAddress [▶ 1477]	Gets the AmsAddress [▶ 648] of the Process Image
	IndexGroup [▶ 1478]	Gets the index group of the Symbol
	IndexOffset [▶ 1478]	Gets the index offset of the Symbol
	InstanceName [▶ 1395]	Gets the name of the instance (without periods (.) (Inherited from Instance [▶ 1388].)
	InstancePath [▶ 1479]	Gets the relative / absolute access path to the instance (with periods (.)) (Overrides Instance.InstancePath [▶ 1396].)
	IsBitType [▶ 1396]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from Instance [▶ 1388].)
	IsByteAligned [▶ 1397]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from Instance [▶ 1388].)
	IsContainerType [▶ 1479]	Gets a value indicating whether the Symbols datatype is a Container type.
	IsDereferencedPointer [▶ 1480]	Gets or sets a value indicating whether an ancestor is a dereferenced Pointer
	IsDereferencedReference [▶ 1481]	Gets or sets a value indicating whether an ancestor is a dereferenced Reference
	IsPersistent [▶ 1397]	Indicates that this instance is persistent. (Inherited from Instance [▶ 1388].)
	IsPointer [▶ 1398]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsPrimitiveType [▶ 1481]	Gets a value indicating whether this instance is primitive.
	IsReadOnly [▶ 1398]	Indicates that this instance is read only. (Inherited from Instance [▶ 1388].)
	IsRecursive [▶ 1482]	Gets a value indicating whether this instance is recursive.

	Name	Description
	IsReference [▶ 1399]	Gets a value indicating whether this instance is reference. (Inherited from Instance [▶ 1388].)
	IsStatic [▶ 1399]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from Instance [▶ 1388].)
	IsTcComInterfacePointer [▶ 1400]	Indicates that this instance is a TcComInterfacePointer. (Inherited from Instance [▶ 1388].)
	IsTypeGuid [▶ 1400]	Indicates that this instance has set TypeGuid flag. (Inherited from Instance [▶ 1388].)
	IsVirtual [▶ 1482]	Gets a value indicating whether this instance is virtual.
	Namespace [▶ 1401]	Gets the namespace name. (Inherited from Instance [▶ 1388].)
	NotificationSettings [▶ 1483]	Gets or sets the notification settings.
	Parent [▶ 1483]	Gets the parent Symbol
	Size [▶ 1401]	Gets the size of the IDataType [▶ 1986] in bytes or Bits dependant on IsBitType [▶ 1396] (Inherited from Instance [▶ 1388].)
	SubSymbolCount [▶ 1484]	Gets the number of SubSymbols
	SubSymbols [▶ 1484]	Gets the SubSymbols of the ISymbol [▶ 2176]
	TypeName [▶ 1402]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from Instance [▶ 1388].)
	ValueEncoding [▶ 1485]	Gets the value encoding.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.1 Symbol.AccessRights Property

Gets the access rights.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolAccessRights AccessRights { get; }
```

Property Value

Type: SymbolAccessRights [▶ 2396]
The access rights.

Implements

IValueSymbol.AccessRights [▶ 2259]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.2 Symbol.Connection Property

Gets the connection that produces values for this IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]
The connection object.

Implements

IValueSymbol.Connection [▶ 2260]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.3 Symbol.ImageBaseAddress Property

Gets the AmsAddress [▶ 648] of the Process Image

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AmsAddress ImageBaseAddress { get; }
```

Property Value

Type: AmsAddress [▶ 648]
The address.

Implements

IAdsSymbol.ImageBaseAddress [▶ 1383]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.4 Symbol.IndexGroup Property

Gets the index group of the Symbol

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint IndexGroup { get; }
```

Property Value

Type: UInt32
The index group.

Implements

IProcessImageAddress.IndexGroup [▶ 2093]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.5 Symbol.IndexOffset Property

Gets the index offset of the Symbol

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint IndexOffset { get; }
```

Property Value

Type: UInt32
The index offset.

Implements

IProcessImageAddress.IndexOffset [▶ 2093]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.6 Symbol.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.))

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Implements

IInstance.InstancePath [▶ 2055]
IInstance.InstancePath [▶ 2055]

Remarks

If this path is relative or absolute depends on the context. IMember [▶ 2065] are using relative paths, ISymbol [▶ 2176]s are using absolute ones.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.7 Symbol.IsContainerType Property

Gets a value indicating whether the Symbols datatype is a Container type.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsContainerType { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

ISymbol.IsContainerType [▶ 2179]

Remarks

Container Types are all types that contain SubElements like

- Array [▶ 1649]
- Pointer [▶ 1649]
- Union [▶ 1649]
- Struct [▶ 1649]
- Function [▶ 1649]
- FunctionBlock [▶ 1649]
- Program [▶ 1649]

and the Alias [▶ 1649] and Reference [▶ 1649] types, if they have a container type as base type.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IDataType.Category [▶ 1988]

6.7.23.1.8 Symbol.IsDereferencedPointer Property

Gets or sets a value indicating whether an ancestor is a dereferenced Pointer

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsDereferencedPointer { get; }
```

Property Value

Type: Boolean
true if this instance is ancestor is pointer; otherwise, false.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.9 Symbol.IsDereferencedReference Property

Gets or sets a value indicating whether an ancestor is a dereferenced Reference

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsDereferencedReference { get; }
```

Property Value

Type: Boolean
true if this instance is ancestor is reference; otherwise, false.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.10 Symbol.IsPrimitiveType Property

Gets a value indicating whether this instance is primitive.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool IsPrimitiveType { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

ISymbol.IsPrimitiveType [▶ 2180]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.11 Symbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsRecursive { get; }
```

Property Value

Type: Boolean
true if this instance is recursive; otherwise, false.

Implements

ISymbol.IsRecursive [▶ 2181]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.12 Symbol.IsVirtual Property

Gets a value indicating whether this instance is virtual.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsVirtual { get; }
```

Property Value

Type: Boolean
true if this instance is virtual; otherwise, false.

Implements

IProcessImageAddress.IsVirtual [▶ 2094]

Remarks

Virtual symbols are are only organizational elements within the Symbols Hierarchy and cannot be accessed seperately by IndexGroup/IndexOffset, Value Read/Writes, notifications or handles.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.13 Symbol.NotificationSettings Property

Gets or sets the notification settings.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public INotificationSettings NotificationSettings { get; set; }
```

Property Value

Type: INotificationSettings [▶ 972]
The notification settings.

Implements

IValueSymbol.NotificationSettings [▶ 2260]

Remarks

The NotificationSettings will be inherited from Parent [▶ 1483] if the setting is not overwritten. If the Root Symbol also doesn't contain the settings, then the DefaultNotificationSettings [▶ 1385] will be returned.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.14 Symbol.Parent Property

Gets the parent Symbol

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Parent { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The parent.

Implements

ISymbol.Parent [▶ 2182]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.15 Symbol.SubSymbolCount Property

Gets the number of SubSymbols

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int SubSymbolCount { get; }
```

Property Value

Type: Int32
The Number of SubSymbols.

Remarks

If the SubSymbols [▶ 1484] collection is not generated yet (WeakReference), then this method is less memory and cpu consuming to use for just determining the the number of child symbols (instead of using SubSymbols.Count)/>

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.16 Symbol.SubSymbols Property

Gets the SubSymbols of the ISymbol [▶ 2176]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolCollection<ISymbol> SubSymbols { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].

Implements

ISymbol.SubSymbols [▶ 2182]

Remarks

Used for Array, Struct, Pointer and Reference instances. Otherwise empty

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.1.17 Symbol.ValueEncoding Property

Gets the value encoding.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual Encoding ValueEncoding { get; }
```

Property Value

Type: Encoding
The value encoding.

Implements

IAttributedInstance.ValueEncoding [▶ 1982]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2 Symbol Methods

The Symbol [▶ 1469] type exposes the following members.

Methods

	Name	Description
	EnsureRights [▶ 1488]	Ensures that the AccessRights are matched.
	Equals [▶ 1489]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode [▶ 1489]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	OnGetSize [▶ 1403]	Handler function getting the size of the Instance [▶ 1388] (Inherited from Instance [▶ 1388].)
	OnReadRawValue [▶ 1490]	Handler function for reading the raw value
	OnReadRawValueAsync [▶ 1490]	Handler function for reading the raw value
	OnReadValue [▶ 1491]	Handler function for reading the dynamic value.
	OnReadValueAsync [▶ 1492]	Handler function for reading the dynamic value.
	OnSetInstanceName [▶ 1492]	Sets a new InstanceName InstancePath (Overrides Instance.OnSetInstanceName(String) [▶ 1404].)
	OnTryReadValue [▶ 1493]	Handler function for reading the dynamic value.
	OnTryWriteValue [▶ 1493]	Handler function for writing the dynamic value
	OnWriteRawValue [▶ 1494]	
	OnWriteRawValueAsync [▶ 1494]	
	OnWriteValue [▶ 1495]	Handler function for writing the dynamic value
	OnWriteValueAsync [▶ 1496]	Handler function for writing the dynamic value
	ReadAnyValue(Type) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadAnyValue(Type, Int32) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadAnyValueAsync [▶ 1498]	Reads the (AnyType) value asynchronously.
	ReadRawValue. [▶ 1499]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1500]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValueAsync [▶ 1501]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.
	ReadValue. [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]

	Name	Description
	ReadValue(Int32) [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValueAsync [▶ 1503]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously.
	SetAttributes [▶ 1404]	Sets the type attributes (Inherited from Instance [▶ 1388].)
	SetContextMask [▶ 1405]	Sets the context mask. (Inherited from Instance [▶ 1388].)
	SetParent [▶ 1504]	Sets the parent symbol.
	ToString [▶ 1505]	Returns a String that represents this instance. (Overrides Instance.ToString. [▶ 1405].)
	TryReadValue [▶ 1505]	Reads the Value of the IValueSymbol [▶ 2254]
	TryWriteValue [▶ 1506]	Writes the Value of the IValueSymbol [▶ 2254]
	UpdateAnyValue(Object.) [▶ 1507]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 1508]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteRawValue(.Byte.) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 1510]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteValue(Object) [▶ 1511]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValue(Object, Int32) [▶ 1512]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValueAsync [▶ 1513]	Writes the Value of the IValueSymbol [▶ 2254]

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.1 Symbol.EnsureRights Method

Ensures that the AccessRights are matched.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected void EnsureRights(
    SymbolAccessRights requested
)
```

Parameters

requested	Type: TwinCAT.TypeSystem.SymbolAccessRights [▶ 2396] The requested rights.

Exceptions

Exception	Condition
InsufficientAccessRightsException [▶ 2076]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.2 Symbol.Equals Method

Equals

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.3 Symbol.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.4 Symbol.OnReadRawValue Method

Handler function for reading the raw value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual byte[] OnReadRawValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32

Return Value

Type: .Byte.
System.Byte[].

Exceptions

Exception	Condition
AdsErrorException [▶ 583]	
AdsErrorException [▶ 583]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.5 Symbol.OnReadRawValueAsync Method

Handler function for reading the raw value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultReadRawAccess> OnReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].
System.Byte[].

Exceptions

Exception	Condition
AdsErrorException [▶ 583]	
AdsErrorException [▶ 583]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.6 Symbol.OnReadValue Method

Handler function for reading the dynamic value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Object OnReadValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32

Return Value

Type: Object
The Value

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.7 Symbol.OnReadValueAsync Method

Handler function for reading the dynamic value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultReadValueAccess> OnReadValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
The Value

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.8 Symbol.OnSetInstanceName Method

Sets a new InstanceName InstancePath

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override void OnSetInstanceName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Instance name.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.9 Symbol.OnTryReadValue Method

Handler function for reading the dynamic value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual int OnTryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout	Type: System.Int32
value	Type: System.Object.

Return Value

Type: Int32
The Value

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.10 Symbol.OnTryWriteValue Method

Handler function for writing the dynamic value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual int OnTryWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout.

Return Value

Type: Int32
System.Int32.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.11 Symbol.OnWriteRawValue Method

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnWriteRawValue(
    ReadOnlyMemory value,
    void timeout
)
```

Parameters

value	Type: ReadOnlyMemory
timeout	Type: System.Void

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.12 Symbol.OnWriteRawValueAsync Method

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultWriteAccess> OnWriteRawValueAsync(
    ReadOnlyMemory value,
    void cancel
)
```

Parameters

value	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWriteAccess [▶ 2575].

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.13 Symbol.OnWriteValue Method

Handler function for writing the dynamic value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.14 Symbol.OnWriteValueAsync Method

Handler function for writing the dynamic value

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultWriteAccess> OnWriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token..

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
Task<ResultWriteAccess>.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.15 Symbol.ReadAnyValue Method

Overload List

	Name	Description
	ReadAnyValue(Type) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type

	Name	Description
	ReadAnyValue(Type, Int32) [▶ 1497]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.ReadAnyValue Method (Type)

Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAnyValue(
    Type managedType
)
```

Parameters

managedType	Type: System.Type The tp.

Return Value

Type: Object
Read value (System.Object).

Implements

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

Reference

Symbol Class [▶ 1469]

ReadAnyValue Overload [▶ 1496]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

Symbol.ReadAnyValue Method (Type, Int32)

Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAnyValue(
    Type managedType,
    int timeout
)
```

Parameters

managedType	Type: System.Type The tp.
timeout	Type: System.Int32 The timeout in ms.

Return Value

Type: Object
Read value (System.Object).

Implements

IValueAnySymbol.ReadAnyValue(Type, Int32) [▶ 2240]

Reference

Symbol Class [▶ 1469]

ReadAnyValue Overload [▶ 1496]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

6.7.23.2.16 Symbol.ReadAnyValueAsync Method

Reads the (AnyType) value asynchronously.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadValueAccess> ReadAnyValueAsync(
    Type managedType,
    CancellationToken cancel
)
```

Parameters

managedType	Type: System.Type Managed type of the value to read.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
A task object that is representing the asynchronous 'ReadAnyValue' operation. The result will be returned in a ResultReadValueAccess [▶ 2566], which contains the Value [▶ 2570] and the ErrorCode [▶ 2559].

Implements

IValueAnySymbol.ReadAnyValueAsync(Type, CancellationToken) [▶ 2241]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.17 Symbol.ReadRawValue Method

Overload List

	Name	Description
	ReadRawValue. [▶ 1499]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1500]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.ReadRawValue Method

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] ReadRawValue()
```

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue. [▶ 2250]

Reference

Symbol Class [▶ 1469]

ReadRawValue Overload [▶ 1499]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.ReadRawValue Method (Int32)

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] ReadRawValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue(Int32) [▶ 2250]

Reference

Symbol Class [▶ 1469]

ReadRawValue Overload [▶ 1499]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.18 Symbol.ReadRawValueAsync Method

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Field Value

Type: Task.ResultReadRawAccess [▶ 2564].
The raw value.

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].
System.Byte[].

Implements

IValueRawSymbol.ReadRawValueAsync(CancellationToken) [▶ 2251]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.19 Symbol.ReadValue Method

Overload List

	Name	Description
	ReadValue. [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValue(Int32) [▶ 1502]	Reads the Value of the IValueSymbol [▶ 2254]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.ReadValue Method

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue()
```

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue. [▶ 2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

Symbol Class [▶ 1469]

ReadValue Overload [▶ 1501]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.ReadValue Method (Int32)

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue(Int32) [▶ 2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [▶ 1469]

ReadValue Overload [▶ 1501]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.20 Symbol.ReadValueAsync Method

Reads the Value of the IValueSymbol [▶ 2254] asynchronously.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultReadValueAccess [▶ 2566] return value and contains the Value [▶ 2570] and the ErrorCode [▶ 2559].

Implements

IValueSymbol.ReadValueAsync(CancellationToken) [▶ 2264]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.21 Symbol.SetParent Method

Sets the parent symbol.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void SetParent(
    ISymbol parent
)
```

Parameters

parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent.

Implements

IHierarchicalSymbol.SetParent(ISymbol) [▶ 2051]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.22 Symbol.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.23 Symbol.TryReadValue Method

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.
value	Type: System.Object. The read value.

Return Value

Type: Int32
The error Code..

Implements

IValueSymbol.TryReadValue(Int32, Object.) [▶ 2265]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.24 Symbol.TryWriteValue Method

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Return Value

Type: Int32
The error code.

Implements

IValueSymbol.TryWriteValue(Object, Int32) [▶ 2265]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.25 Symbol.UpdateAnyValue Method

Overload List

	Name	Description
	UpdateAnyValue(Object.) [▶ 1507]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 1508]	Reads the value of this Value [▶ 2254] into the specified managed value.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.UpdateAnyValue Method (Object.)

Reads the value of this Value [▶ 2254] into the specified managed value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void UpdateAnyValue(
    ref Object managedObject
)
```

Parameters

managedObject	Type: System.Object. The managed object.

Return Value

Type:
Read value (System.Object).

Implements

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

Reference

Symbol Class [▶ 1469]

UpdateAnyValue Overload [▶ 1507]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

Symbol.UpdateAnyValue Method (Object., Int32)

Reads the value of this Value [▶ 2254] into the specified managed value.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void UpdateAnyValue(
    ref Object managedObject,
    int timeout
)
```

Parameters

managedObject	Type: System.Object. The managed object.
timeout	Type: System.Int32 The timeout.

Return Value

Type:
Read value (System.Object).

Implements

IValueAnySymbol.UpdateAnyValue(Object., Int32) [▶ 2242]

Reference

Symbol Class [▶ 1469]

UpdateAnyValue Overload [▶ 1507]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

6.7.23.2.26 Symbol.WriteRawValue Method

Overload List

	Name	Description
	WriteRawValue(.Byte.) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

	Name	Description
	WriteRawValue(.Byte., Int32) [▶ 1509]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.WriteRawValue Method (.Byte.)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteRawValue(
    byte[] rawValue
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.

Field Value

Type:
The value.

Implements

IValueRawSymbol.WriteRawValue(.Byte.) [▶ 2252]

Reference

Symbol Class [▶ 1469]

WriteRawValue Overload [▶ 1508]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.WriteRawValue Method (.Byte., Int32)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

rawValue	Type: .System.Byte. The raw value.
timeout	Type: System.Int32 The timeout.

Field Value

Type:
The raw value.

Implements

IValueRawSymbol.WriteRawValue(.Byte., Int32) [▶ 2252]

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [▶ 1469]

WriteRawValue Overload [▶ 1508]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.27 Symbol.WriteRawValueAsync Method

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A task that represents the asynchronous read operation. The ResultRead [▶ 1008] parameter contains the total number of bytes read into the buffer (ReadBytes [▶ 1010]) and the ErrorCode [▶ 992] after execution..

Implements

IValueRawSymbol.WriteRawValueAsync(.Byte., CancellationToken) [▶ 2253]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.28 Symbol.WriteValue Method

Overload List

	Name	Description
	WriteValue(Object) [▶ 1511]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValue(Object, Int32) [▶ 1512]	Writes the Value of the IValueSymbol [▶ 2254]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.WriteValue Method (Object)

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    Object value
)
```

Parameters

value	Type: System.Object The value.

Implements

IValueSymbol.WriteValue(Object) [▶ 2266]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

Symbol Class [▶ 1469]

WriteValue Overload [▶ 1511]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

Symbol.WriteValue Method (Object, Int32)

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Implements

IValueSymbol.WriteValue(Object, Int32) [▶ 2267]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

Symbol Class [▶ 1469]

WriteValue Overload [▶ 1511]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.2.29 Symbol.WriteValueAsync Method

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultWriteAccess [▶ 2575] return value and contains the ErrorCode [▶ 2559].

Implements

IValueSymbol.WriteValueAsync(Object, CancellationToken) [▶ 2268]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.3 Symbol Events

The Symbol [▶ 1469] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1514]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.
	ValueChanged [▶ 1514]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.3.1 Symbol.RawValueChanged Event

Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler.RawValueChangedEventArgs [▶ 2289].

Implements

IValueRawSymbol.RawValueChanged [▶ 2254]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.3.2 Symbol.ValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

Value

Type: System.EventHandler.ValueChangedEventArgs [▶ 2439].

Implements

IValueSymbol.ValueChanged [▶ 2269]

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.4 Symbol Operators

The Symbol [▶ 1469] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 1515]	Operator==
	Inequality [▶ 1516]	Implements the != operator.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.4.1 Symbol.Equality Operator

Operator==

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    Symbol o1,
    Symbol o2
)
```

Parameters

o1	Type: TwinCAT.Ads.TypeSystem.Symbol [▶ 1469] The o1.

o2	Type: TwinCAT.Ads.TypeSystem.Symbol [▶ 1469] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.4.2 Symbol.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    Symbol o1,
    Symbol o2
)
```

Parameters

o1	Type: TwinCAT.Ads.TypeSystem.Symbol [▶ 1469] The o1.
o2	Type: TwinCAT.Ads.TypeSystem.Symbol [▶ 1469] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.5 Symbol Fields

The Symbol [▶ 1469] type exposes the following members.

Fields

	Name	Description
	syncObject [▶ 1517]	Synchronization object

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.23.5.1 Symbol.syncObject Field

Synchronization object

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Object syncObject
```

Field Value

Type: Object

Reference

Symbol Class [▶ 1469]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24 SymbolIterator Class

Iterator class for enumerations of Symbols [▶ 2176].

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.SymbolIterator [▶ 2532].ISymbol [▶ 2176].
    TwinCAT.Ads.TypeSystem.SymbolIterator
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SymbolIterator : SymbolIterator<ISymbol>
```

The SymbolIterator type exposes the following members.

Constructors

	Name	Description
	SymbolIterator(IInstanceCollection.ISymbol.) [▶ 1520]	Initializes a new instance of the SymbolIterator class.

	Name	Description
	SymbolIterator(IInstanceCollection.ISymbol., Func.ISymbol, Boolean.) [▶ 1520]	Initializes a new instance of the SymbolIterator class.
	SymbolIterator(IEnumerable.ISymbol., Boolean) [▶ 1521]	Initializes a new instance of the SymbolIterator class.
	SymbolIterator(IEnumerable.ISymbol., Boolean, Func.ISymbol, Boolean.) [▶ 1521]	Initializes a new instance of the SymbolIterator class.

Properties

	Name	Description
	Mask [▶ 2538]	Gets or sets the SymbolIterationMask [▶ 2532] (Inherited from SymbolIterator.T. [▶ 2532].)
	SymbolRecursionDetection [▶ 2538]	Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default). (Inherited from SymbolIterator.T. [▶ 2532].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2539]	Gets the enumerator that enumerates through a collection (Inherited from SymbolIterator.T. [▶ 2532].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This iterator class can be used to iterate over collections of symbol trees (root symbols + sub symbols). By constructor the user can choose if the iterator works recursively within the symbol tree and optionally a filter function to select only specific symbols (predicate).

Examples

The following example shows how to determine, browse and filter symbols.

Browsing and filtering Symbols

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols  = await loader.GetSymbolsAsync(cancel);

    if (resultSymbols.Succeeded)
    {
    Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
    ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
        valueToRead = (uint)resultRead.Value;

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
    }
}
```

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.1 SymbolIterator Constructor

Overload List

	Name	Description
	SymbolIterator(IInstanceCollection.ISymbol.) [▶ 1520]	Initializes a new instance of the SymbolIterator [▶ 1517] class.
	SymbolIterator(IInstanceCollection.ISymbol., Func.ISymbol, Boolean.) [▶ 1520]	Initializes a new instance of the SymbolIterator [▶ 1517] class.
	SymbolIterator(IEnumerable.ISymbol., Boolean) [▶ 1521]	Initializes a new instance of the SymbolIterator [▶ 1517] class.

	Name	Description
	SymbolIterator(IEnumerable.ISymbol., Boolean, Func.ISymbol, Boolean.) [▶ 1521]	Initializes a new instance of the SymbolIterator [▶ 1517] class.

Reference

SymbolIterator Class [▶ 1517]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.1.1 SymbolIterator Constructor (IInstanceCollection.ISymbol.)

Initializes a new instance of the SymbolIterator [▶ 1517] class.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IInstanceCollection<ISymbol> symbols
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].ISymbol [▶ 2176]. The symbol collection

Reference

SymbolIterator Class [▶ 1517]

SymbolIterator Overload [▶ 1519]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.1.2 SymbolIterator Constructor (IInstanceCollection.ISymbol., Func.ISymbol, Boolean.)

Initializes a new instance of the SymbolIterator [▶ 1517] class.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IInstanceCollection<ISymbol> symbols,
    Func<ISymbol, bool> predicate
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].ISymbol [▶ 2176]. The symbol collection.
predicate	Type: System.Func.ISymbol [▶ 2176], Boolean. The predicate / filter function

Reference

SymbolIterator Class [▶ 1517]

SymbolIterator Overload [▶ 1519]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.1.3 SymbolIterator Constructor (IEnumerable.ISymbol., Boolean)

Initializes a new instance of the SymbolIterator [▶ 1517] class.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IEnumerable<ISymbol> symbols,
    bool recurse
)
```

Parameters

symbols	Type: System.Collections.Generic.IEnumerable.ISymbol [▶ 2176]. The symbol enumeration.
recurse	Type: System.Boolean if set to true, the iterator works recursively over all subsymbols.

Reference

SymbolIterator Class [▶ 1517]

SymbolIterator Overload [▶ 1519]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.1.4 SymbolIterator Constructor (IEnumerable.ISymbol., Boolean, Func.ISymbol, Boolean.)

Initializes a new instance of the SymbolIterator [▶ 1517] class.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IEnumerable<ISymbol> symbols,
    bool recurse,
    Func<ISymbol, bool> predicate
)
```

Parameters

symbols	Type: System.Collections.Generic.IEnumerable.ISymbol [▶ 2176]. The symbol collection.
recurse	Type: System.Boolean if set to true, the iterator works recursively over all subsymbols.
predicate	Type: System.Func.ISymbol [▶ 2176], Boolean. The predicate / filter function.

Reference

SymbolIterator Class [▶ 1517]

SymbolIterator Overload [▶ 1519]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.2 SymbolIterator Properties

The SymbolIterator [▶ 1517] type exposes the following members.

Properties

	Name	Description
	Mask [▶ 2538]	Gets or sets the SymbolIterationMask [▶ 2532] (Inherited from SymbolIterator.T. [▶ 2532].)
	SymbolRecursionDetection [▶ 2538]	Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default). (Inherited from SymbolIterator.T. [▶ 2532].)

Reference

SymbolIterator Class [▶ 1517]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.24.3 SymbolIterator Methods

The SymbolIterator [▶ 1517] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2539]	Gets the enumerator that enumerates through a collection (Inherited from SymbolIterator.T. [▶ 2532].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SymbolIterator Class [▶ 1517]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.25 SymbolLoaderFactory Class

The class SymbolLoaderFactory is used to create a new instance of the AdsSymbolLoader initialized to the parametrized mode (SymbolBrowser V2, new Version)

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.SymbolLoaderFactory
Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class SymbolLoaderFactory
```

The SymbolLoaderFactory type exposes the following members.

Methods

	Name	Description
	Create [▶ 1524]	Creates the specified connection.

Remarks

The Symbol Loader (V2) supports the following modes [▶ 149]. Flat [▶ 149]The flat mode organizes the Symbols in a flat list. This mode is available in all .NET versions. VirtualTree [▶ 149]The virtual tree mode organizes the Symbols hierarchically with parent-child relationships. This mode is available in all .NET Versions. DynamicTree [▶ 149]The Dynamic tree mode organizes the Symbols hierarchically and (dynamically) creates struct members, array elements and enum fields on the fly. This feature is only available on platforms that support the Dynamic Language Runtime (DLR), actually all .NET Framework Version larger than 4.0. Virtual instances means, that all Symbols are ordered within a tree structure. For that symbol nodes that are not located on a fixed address, a Virtual Symbol will be created. Setting the virtualInstance parameter to 'false' means, that the located symbols will be returned in a flattened list.

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

TwinCAT.SymbolLoaderSettings [▶ 140]

6.7.25.1 SymbolLoaderFactory Methods

The SymbolLoaderFactory [▶ 1523] type exposes the following members.

Methods

	Name	Description
	Create [▶ 1524]	Creates the specified connection.

Reference

SymbolLoaderFactory Class [▶ 1523]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.25.1.1 SymbolLoaderFactory.Create Method

Creates the specified connection.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ISymbolLoader Create(
    IConnection connection,
    ISymbolLoaderSettings settings
)
```

Parameters

connection	Type: TwinCAT.IConnection [▶ 74] The connection.

settings	Type: TwinCAT.ISymbolLoaderSettings [▶ 99] The settings.

Return Value

Type: ISymbolLoader [▶ 2200]
ISymbolLoader.

Examples

The following sample shows how to create a dynamic version of the SymbolLoader V2. The dynamic symbol loader makes use of the Dynamic Language Runtime (DLR) of the .NET Framework. That means Structures, Arrays and Enumeration types and instances are generated 'on-the-fly' during symbol Browsing. These created dynamic objects are a one to one representation of the Symbol Server target objects (e.g the IEC61131 types on the PLC). Dynamic language features are only available from .NET4 upwards.

Dynamic Tree Mode

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;
```

```
        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;


        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
```

```
        }
    }

    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
f"));
        }
    }
    }
}
```

The following sample shows how to create a static (non dynamic) version of the SymbolLoader V2. The static symbol loader in version 2 is a nearly code compatible version of the Dynamic Loader, only the dynamic creation of objects is not available. The reason for supporting this mode is that .NET Framework Versions lower than Version 4.0 (CLR2) doesn't support the Dynamic Language Runtime (DLR). The SymbolLoader V2 static object is supported from .NET 2.0 on.

Virtual Tree Mode

```
using System;
using System.Threading;
using System.Diagnostics;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.Ads.TypeSystem;

namespace Sample
{
    class SymbolBrowserProgramV2VirtualTree
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        ConsoleLogger logger = new ConsoleLogger();

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for start:");
        Console.ReadLine();

        //logger.Active = false;

        Stopwatch stopper = new Stopwatch();

        // Parse the command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        stopper.Start();

        using (AdsClient client = new AdsClient())
        {
        //client.Synchronize = false;

        // Connect the AdsClient to the device target.
        client.Connect(address);
```

```
        // Creates the Symbol Objects as hierarchical tree
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree, ValueA
ccessMode.IndexGroupOffsetPreferred);
        ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

        // Dump Datatypes from Target Device
        Console.WriteLine(string.Format("Dumping '{0}' DataTypes:", symbolLoader.DataTypes.Count));
        foreach (IDataType type in symbolLoader.DataTypes)
        {
            logger.DumpType(type);
        }
        Console.WriteLine("");

        // Dump Symbols from target device
        Console.WriteLine("Dumping '{0}' Symbols:", symbolLoader.Symbols.Count);
        foreach (ISymbol symbol in symbolLoader.Symbols)
        {
            logger.DumpSymbol(symbol,0);
        }
        }
        stopper.Stop();
        TimeSpan elapsed = stopper.Elapsed;

        Console.WriteLine("");
        Console.WriteLine("Browsing complete tree: {0},
({1} DataTypes, {2} Symbols)",elapsed,logger.DataTypesCount,logger.DataTypesCount);
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }
```

Examples

The SymbolLoader V2 static object is supported from .NET 2.0 on.

Flat Mode

```
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SymbolBrowserProgramV2Flat
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        ConsoleLogger logger = new ConsoleLogger();

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for start:");
        Console.ReadLine();

        //logger.Active = false;

        Stopwatch stopper = new Stopwatch();

        // Parse the command line arguments
        AmsAddress address = ArgParser.Parse(args);

        stopper.Start();

        // Create the ADS Client
        using (AdsClient client = new AdsClient())
        {
        //client.Synchronize = false;

        // Connect to Address
        client.Timeout = 30000;
```

```
        client.Connect(address);

        // Creates the Symbol Objects in Flat Mode (Flat list)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.Flat, ValueAccessMo
de.IndexGroupOffsetPreferred);
        ISymbolLoader symbolLoader = SymbolLoaderFactory.Create(client, settings);

        // Dump Datatypes from Target Device
        Console.WriteLine(string.Format("Dumping '{0}' DataTypes:",symbolLoader.DataTypes.Count));
        foreach (IDataType type in symbolLoader.DataTypes)
        {
            logger.DumpType(type);
        }

        Console.WriteLine("");

        // Dump Symbols from target device
        Console.WriteLine("Dumping '{0}' Symbols:",symbolLoader.Symbols.Count);
        foreach (ISymbol symbol in symbolLoader.Symbols)
        {
            logger.DumpSymbol(symbol,0);
        }
        }
        stopper.Stop();
        TimeSpan elapsed = stopper.Elapsed;

        Console.WriteLine("");
        Console.WriteLine("Browsing complete tree: {0},
({1} DataTypes, {2} Symbols)", elapsed, logger.DataTypesCount, logger.DataTypesCount);
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }
```

Examples

Argument Parser

```
public static class ArgParser
{
    /// <summary>
    /// Parses the arguments.
    /// </summary>
    /// <param name="args">The arguments.</param>
    /// <returns>AmsAddress.</returns>
    public static AmsAddress Parse(string[] args)
    {
    AmsNetId netId = AmsNetId.Local;
    int port = 851;

    if (args != null)
    {
        if (args.Length > 0 && args[0] != null)
        netId = AmsNetId.Parse(args[0]);

        if (args.Length > 1 && args[1] != null)
        port = int.Parse(args[1]);
    }
    return new AmsAddress(netId, port);
    }
}
```

Dumping Symbols

```
/// <summary>
/// Console logger
/// </summary>
public class ConsoleLogger
{
    public ConsoleLogger()
    {
    }
    bool _active = true;

    /// <summary>
    /// Gets or sets a value indicating whether this ConsoleLogger is active.
    /// </summary>
    /// <value><c>true</c> if active; otherwise, <c>false</c>.</value>
    public bool Active
```

```
    {
    get { return _active; }
    set
    {
        _active = value;
    }
    }

    int _dataTypes = 0;

    /// <summary>
    /// Gets the number of dumped dataTypes.
    /// </summary>
    /// <value>The data types count.</value>
    public int DataTypesCount
    {
    get { return _dataTypes; }
    }

    int _symbols = 0;

    /// <summary>
    /// Gets the number of dumped symbols
    /// </summary>
    /// <value>The symbols count.</value>
    public int SymbolsCount
    {
    get { return _symbols; }
    }


    /// <summary>
    /// Dumps the data type.
    /// </summary>
    /// <param name="dataType">Data Type.</param>
    public void DumpType(IDataType dataType)
    {
    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType.Categ
ory, dataType.Size));

    switch (dataType.Category)
    {
        case DataTypeCategory.Alias:
        IAliasType alias = (IAliasType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Alias BaseType: {0}", alias.BaseTypeName));
        break;

        case DataTypeCategory.Enum:

        //IEnumType<ushort> enumType = (IEnumType<ushort>)dataType;
        IEnumType enumType = (IEnumType)dataType;
        WriteLine(GetPrefix(1) + string.Format("Enum BaseType: {0}", enumType.BaseTypeName));

        foreach (IEnumValue enumValue in enumType.EnumValues)
        {
            WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumValu
e.Primitive));
        }
        break;
        case DataTypeCategory.Array:

        IArrayType arrayType = (IArrayType)dataType;
        int i = 0;

        foreach (IDimension dim in arrayType.Dimensions)
        {
            WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}", i+
+, dim.LowerBound, dim.ElementCount));
        }
        break;
        case DataTypeCategory.Struct:
        IStructType structType = (IStructType)dataType;

        foreach (IMember member in structType.Members)
        {
            WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}", member.Offset
, member.InstanceName, member.TypeName));
        }
        break;
        default:
```

```
        break;
    }

    foreach (ITypeAttribute attribute in dataType.Attributes)
    {
        WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value));
    }
    if (!string.IsNullOrEmpty(dataType.Comment))
    {
        WriteLine(GetPrefix(1) + string.Format("Comment: {0}", dataType.Comment));
    }

    IRpcCallableType rpcCallable = dataType as IRpcCallableType;

    if (rpcCallable != null)
    {
        foreach (IRpcMethod rpcMethod in rpcCallable.RpcMethods)
        {
        if (string.IsNullOrEmpty(rpcMethod.Comment))
            WriteLine(GetPrefix(1) + string.Format("Method: {0}", rpcMethod));
        else
            WriteLine(GetPrefix(1) + string.Format("Method: {0}, Comment: {1}", rpcMethod, rpcMethod
.Comment));
        }
    }
    _dataTypes++;
    }

    ///// <summary>
    ///// Dumps the Datatype to Console
    ///// </summary>
    ///// <param name="dataType">DataType.</param>
    //public void DumpType(ITcAdsDataType dataType)
    //{
    //    // Dump the Attributes (PLC Metadata)
    //    foreach (ITypeAttribute attribute in dataType.Attributes)
    //    {
    //    WriteLine(GetPrefix(1) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Value)
);
    //    }

    //    WriteLine(string.Format("DataType: {0}, Category: {1}, Size: {2}", dataType.Name, dataType
.Category, dataType.Size));

    //    if (dataType.BaseType != null)
    //    {
    //    WriteLine(GetPrefix(1) + string.Format("BaseType: {0}", dataType.BaseType));
    //    }

    //    switch (dataType.Category)
    //    {
    //    case DataTypeCategory.Enum:
    //        foreach (IEnumValue enumValue in dataType.EnumValues)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("Name: {0}, Value: {1}", enumValue.Name, enumVa
lue.Primitive));
    //        }
    //        break;
    //    case DataTypeCategory.Array:
    //        int i = 0;
    //        foreach (IDimension dim in dataType.Dimensions)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("{0}: LowerBound: {1}, Elements: {2}", i+
+, dim.LowerBound, dim.ElementCount));
    //        }
    //        break;
    //    case DataTypeCategory.Struct:
    //        foreach (ITcAdsSubItem subItem in dataType.SubItems)
    //        {
    //        WriteLine(GetPrefix(2) + string.Format("Offset {0}: Name: {1}, Type: {2}", subItem.Off
set, subItem.SubItemName, subItem.Name));
    //        }
    //        break;
    //    default:
    //        break;
    //    }
    //    _dataTypes++;
    //}

    /// <summary>
```

```
    /// Dump Symbol
    /// </summary>
    /// <param name="symbol">The symbol.</param>
    /// <param name="level">Output indentation level</param>
    public void DumpSymbol(ISymbol symbol, int level)
    {
    IDataType type = symbol.DataType as IDataType;

    foreach (ITypeAttribute attribute in symbol.Attributes)
    {
        WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Valu
e));
    }

    WriteLine(GetPrefix(level) + string.Format("{0} : {1} (IG: 0x{2} IO: 0x{3} size:
{4})", symbol.InstanceName, symbol.TypeName, ((IAdsSymbol)symbol).IndexGroup.ToString("x"), ((IAdsSy
mbol)symbol).IndexOffset.ToString("x"), symbol.Size));

    if (symbol.Category == DataTypeCategory.Array)
    {
        IArrayInstance arrInstance = (IArrayInstance)symbol;
        IArrayType arrType = (IArrayType)symbol.DataType;

        int count = 0;
        level++;

        foreach (ISymbol arrayElement in arrInstance.Elements)
        {
        DumpSymbol(arrayElement, level);
        count++;

        if (count > 20) // Write only the first 20 to limit output
            break;
        }
    }
    else if (symbol.Category == DataTypeCategory.Struct)
    {
        IStructInstance structInstance = (IStructInstance)symbol;
        IStructType structType = (IStructType)symbol.DataType;

        level++;

        foreach (ISymbol member in structInstance.MemberInstances)
        {
        DumpSymbol(member, level);
        }
    }
    _symbols++;
    }

    ///// <summary>
    ///// Dumps the specified Symbol to the Console
    ///// </summary>
    ///// <param name="symbol">The symbol.</param>
    ///// <param name="level">The level.</param>
    //public void DumpSymbol(IAdsSymbol2 symbol, int level)
    //{
    //    // Dump Attributes of the Symbol
    //    foreach (ITypeAttribute attribute in symbol.Attributes)
    //    {
    //    WriteLine(GetPrefix(level) + string.Format("{{ {0} : {1} }}", attribute.Name, attribute.Va
lue));
    //    }

    //    ITcAdsSymbolBrowser subSymbolProvider = (ITcAdsSymbolBrowser)symbol;

    //    // Dump The Symbol
    //    WriteLine(GetPrefix(level) + string.Format("{0} : {1} ({2}, IG: 0x{3} IO: 0x{4} size:
{6} subCount:
{5})", symbol.Name, symbol.TypeName, symbol.DataTypeId, symbol.IndexGroup.ToString("x"), symbol.Inde
xOffset.ToString("x"), subSymbolProvider.SubSymbols.Count, symbol.Size));
    //    level++;

    //    // Dump all SubSymbols with indentation
    //    foreach (IAdsSymbol2 subSymbol in ((ITcAdsSymbolBrowser)symbol).SubSymbols)
    //    {
    //    DumpSymbol(subSymbol, level);
    //    }
    //    _symbols++;
    //}
```

```
    /// <summary>
    /// Dump namespace.
    /// </summary>
    /// <param name="ns">The namespace.</param>
    public void DumpNamespace(INamespace<IDataType> ns)
    {
    WriteLine("Namespace: {0}, DataTypes: {1}", ns.Name, ns.DataTypes.Count);

    foreach (IDataType type in ns.DataTypes)
    {
        DumpType(type);
    }
    }

    /// <summary>
    /// Get the indentation prefix
    /// </summary>
    /// <param name="level">The level.</param>
    /// <returns>System.String.</returns>
    public string GetPrefix(int level)
    {
    return "".PadLeft(level * 3);
    }

    /// <summary>
    /// Writes a line to the Console
    /// </summary>
    /// <param name="message">The message.</param>
    public void WriteLine(string message)
    {
    if (Active)
    {
        Console.WriteLine(message);
    }
    }

    /// <summary>
    /// Writes a line to the console
    /// </summary>
    /// <param name="format">The format.</param>
    /// <param name="args">The arguments.</param>
    public void WriteLine(string format, params object[] args)
    {
    if (Active)
    {
        Console.WriteLine(format, args);
    }
    }
}
```

Reference

SymbolLoaderFactory Class [▶ 1523]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.26 UnionType Class

Represents a union type

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.UnionType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class UnionType : DataType,
    IUnionType, IDataType, IBitSize
```

The UnionType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Fields [▶ 1536]	Gets a read only collection of the Fields [▶ 2040] of the IUnionType [▶ 2224].
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.26.1 UnionType Properties

The UnionType [▶ 1533] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Fields [▶ 1536]	Gets a read only collection of the Fields [▶ 2040] of the IUnionType [▶ 2224].
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)

	Name	Description
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

UnionType Class [▶ 1533]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.26.1.1 UnionType.Fields Property

Gets a read only collection of the Fields [▶ 2040] of the IUnionType [▶ 2224].

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IFieldCollection Fields { get; }
```

Property Value

Type: IFieldCollection [▶ 2042]
The members as read only collection.

Implements

IUnionType.Fields [▶ 2226]

Reference

UnionType Class [▶ 1533]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.26.2 UnionType Methods

The UnionType [▶ 1533] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1363]	Returns a String that represents this instance. (Inherited from DataType [▶ 1349].)

Reference

UnionType Class [▶ 1533]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27 WStringType Class

Represents an Unicode string (Wide string)

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.TypeSystem.DataType [▶ 1349]
    TwinCAT.Ads.TypeSystem.WStringType

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class WStringType : DataType,
    IStringType, IDataType, IBitSize
```

The WStringType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Encoding [▶ 1539]	Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsFixedLength [▶ 1540]	Gets a value indicating whether the string is of fixed length.

	Name	Description
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	Length [▶ 1540]	Gets the number of characters within the string.
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1541]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.1 WStringType Properties

The WStringType [▶ 1537] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1352]	Gets the attributes of the IDataType [▶ 1986] (Inherited from DataType [▶ 1349].)
	BitSize [▶ 1353]	Gets the size of the DataType [▶ 1349] in bits. (Inherited from DataType [▶ 1349].)
	ByteSize [▶ 1353]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DataType [▶ 1349].)
	Category [▶ 1354]	Gets the Data Type category (Inherited from DataType [▶ 1349].)
	Comment [▶ 1354]	Gets the comment. (Inherited from DataType [▶ 1349].)
	Encoding [▶ 1539]	Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))

	Name	Description
	FullName [▶ 1355]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from DataType [▶ 1349].)
	Id [▶ 1355]	Gets the ID of the DataType (Inherited from DataType [▶ 1349].)
	IsBitType [▶ 1356]	Gets a value indicating whether this IDataType [▶ 1986] is a bit mapping Type (Inherited from DataType [▶ 1349].)
	IsByteAligned [▶ 1356]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DataType [▶ 1349].)
	IsContainer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from DataType [▶ 1349].)
	IsFixedLength [▶ 1540]	Gets a value indicating whether the string is of fixed length.
	IsPointer [▶ 1357]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from DataType [▶ 1349].)
	IsPrimitive [▶ 1358]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from DataType [▶ 1349].)
	IsReference [▶ 1359]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from DataType [▶ 1349].)
	Length [▶ 1540]	Gets the number of characters within the string.
	ManagedType [▶ 1359]	Gets the corresponding .NET Type if attached. (Inherited from DataType [▶ 1349].)
	Name [▶ 1360]	Gets the name of the Data Type (without namespace) (Inherited from DataType [▶ 1349].)
	Namespace [▶ 1360]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from DataType [▶ 1349].)
	Size [▶ 1361]	Gets the Size of the DataType [▶ 1349] in Bytes or bits. (Inherited from DataType [▶ 1349].)

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.1.1 WStringType.Encoding Property

Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding Encoding { get; }
```

Property Value

Type: Encoding
The encoding.

Implements

IStringType.Encoding [▶ 2157]

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.1.2 WStringType.IsFixedLength Property

Gets a value indicating whether the string is of fixed length.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsFixedLength { get; }
```

Property Value

Type: Boolean
true if this instance is fixed length; otherwise, false.

Implements

IStringType.IsFixedLength [▶ 2157]

Exceptions

Exception	Condition
NotImplementedException	

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.1.3 WStringType.Length Property

Gets the number of characters within the string.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Length { get; }
```

Property Value

Type: Int32
The length.

Implements

IStringType.Length [▶ 2158]

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.2 WStringType Methods

The WStringType [▶ 1537] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	ToString [▶ 1541]	Returns a String that represents this instance. (Overrides DataType.ToString. [▶ 1363].)

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.7.27.2.1 WStringType.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.Ads.TypeSystem [▶ 1328]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

WStringType Class [▶ 1537]

TwinCAT.Ads.TypeSystem Namespace [▶ 1328]

6.8 TwinCAT.Ads.ValueAccess Namespace

Root namespace for ADS value access.

Enumerations

	Enumeration	Description
	ValueAccessMode [▶ 1542]	Enum ValueAccessMethod

6.8.1 ValueAccessMode Enumeration

Enum ValueAccessMethod

Namespace: TwinCAT.Ads.ValueAccess [▶ 1542]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum ValueAccessMode
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	IndexGroupOffset	1	Value access via Index Group and Offset Only
	Symbolic	2	Symbolic access via Instance Path only.
	IndexGroupOffsetPreferred	3	Uses IndexGroup IndexOffset Preferred (and Symbolic for Dereferenced Pointers / References)
	Default	2	The Default access mode (Symbolic)

Remarks

Mode	Description
None	None/Uninitialized. No Valid mode.

Mode	Description
IndexGroupOffset	Communicates over IndexGroup/IndexOffset only. This is the most direct/ efficient access into the Process image. The advantage is that, the symbol access is done via 1 ADS round trip. Disadvantages are that not all Symbols can be accessed via IG/IO (e.g. References) and IndexOffsets could be invalid after online changes / PlcProgram downloads. Detection of these events and following invalidation of all changed symbols need to be done within the user application.
Symbolic	The Symbolic-only mode is the most safe mode to use but needs more time than the IndexGroupOffset. It could need up to 3 ADS round trips (create handle, access value, close handle) but is not influenced by online changes or / plcProgram downloads.
IndexGroupOffsetPreferred	This is a mixed access mode. For symbols, where it is possible it uses the IndexGroup/IndexOffset. For others it chooses the Symbolic access.
Default	The Default-Mode setting if no other ValueAccessMode is specified. This is set to Symbolic.

Reference

TwinCAT.Ads.ValueAccess Namespace [▶ 1542]

6.9 TwinCAT.Ams Namespace

Classes

	Class	Description
	AmsConfiguration [▶ 1543]	Static configuration of the Ams Router system.
	AmsServerException [▶ 1548]	Ams Server Exception class Implements the Exception

Enumerations

	Enumeration	Description
	AmsServerErrorCode [▶ 1548]	Ams Server Error Codes.

6.9.1 AmsConfiguration Class

Static configuration of the Ams Router system.

Inheritance Hierarchy

System.Object
  TwinCAT.Ams.AmsConfiguration
Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static class AmsConfiguration
```

The AmsConfiguration type exposes the following members.

Properties

	Name	Description
	DefaultRouterEndPoint [▶ 1545]	Gets the default router end point (IPAddress.Loopback, Port 0xBF02)
	RouterAddress [▶ 1545]	Gets the actually configured router address.
	RouterEndPoint [▶ 1546]	Gets or sets the default router IPEndPoint.
	RouterPort [▶ 1547]	Gets actually configured router port.

Fields

	Name	Description
	DEFAULT_TCP_PORT [▶ 1547]	The default TCP port (0xBF02, 48898)

Remarks

If the router is intended to run seperately of the (virtual) system that instantiates the AdsClient or AdsServer (e.g. in UnitTests), some global static settings must be adopted to enable different network scenarious. In the default case the AdsServers and AdsClients uses the Loopback TCP port **0xBF02** to communicate internally. If the systems of AdsServer/AdsClient is split from the router (e.g. running the router isolated in a virtual machine), the Router Endpoint must be set via RouterEndPoint [▶ 1546] setter.

Reference

TwinCAT.Ams Namespace [▶ 1543]

6.9.1.1 AmsConfiguration Properties

The AmsConfiguration [▶ 1543] type exposes the following members.

Properties

	Name	Description
	DefaultRouterEndPoint [▶ 1545]	Gets the default router end point (IPAddress.Loopback, Port 0xBF02)

	Name	Description
	RouterAddress [▶ 1545]	Gets the actually configured router address.
	RouterEndPoint [▶ 1546]	Gets or sets the default router IPEndPoint.
	RouterPort [▶ 1547]	Gets actually configured router port.

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

6.9.1.1.1 AmsConfiguration.DefaultRouterEndPoint Property

Gets the default router end point (IPAddress.Loopback, Port 0xBF02)

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IPEndPoint DefaultRouterEndPoint { get; }
```

Property Value

Type: IPEndPoint
The default router end point.

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

6.9.1.1.2 AmsConfiguration.RouterAddress Property

Gets the actually configured router address.

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IPAddress RouterAddress { get; }
```

Property Value

Type: IPAddress
The router address.

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

AmsConfiguration.RouterEndPoint [▶ 1546]

6.9.1.1.3 AmsConfiguration.RouterEndPoint Property

Gets or sets the default router IPEndPoint.

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static IPEndPoint RouterEndPoint { get; set; }
```

Property Value

Type: IPEndPoint
The IPEndPoint used by Ams Clients/Servers.

Remarks

This setting allows to change the Router Endpoint - which listens by default on **IPAddress 127.0.0.1, Port 0xBF02**. An application for this could be to move the router to a different (virtual) system, separated from the system running the **AdsServer** or **AdsClient**. This endpoint can only be changed process wide and should be done before the first access to the Router occurs (e.g. getting the local NetId).

ATTENTION: Bending this Endpoint is not possible with the standard TwinCAT Router. As security feature the TwinCAT Router only accepts Loopback connections to 127.0.0.1, connection requests from the outside will be closed immediately.

The simplistic .NET Core Router implementation in class 'TwinCAT.Ads.AdsRouter.AmsTcpIpRouter' from nuget package 'Beckhoff.TwinCAT.Ads.TcpRouter' or the 'Beckhoff.TwinCAT.Ads.AdsRouterConsole' doesn't have that restriction and can be used more flexible for use with virtual environments like VirtualMachines or Docker. flexible.

The following sample demonstrates how to configure the (internal) RouterEndPoint to IPAddress 1.2.3.4:42 . This must be done before AdsServer or AdsClient are instantiated.

C#

```
AmsConfiguration.RouterEndPoint = new IPEndpoint(IPAddress.Parse("1.2.3.4"),42);
```

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

6.9.1.1.4 AmsConfiguration.RouterPort Property

Gets actually configured router port.

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static int RouterPort { get; }
```

Property Value

Type: Int32
The router port.

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

AmsConfiguration.RouterEndPoint [▶ 1546]

6.9.1.2 AmsConfiguration Fields

The AmsConfiguration [▶ 1543] type exposes the following members.

Fields

	Name	Description
	DEFAULT_TCP_PORT [▶ 1547]	The default TCP port (0xBF02, 48898)

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

6.9.1.2.1 AmsConfiguration.DEFAULT_TCP_PORT Field

The default TCP port (0xBF02, 48898)

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public const int DEFAULT_TCP_PORT = 48898
```

Field Value

Type: Int32

Reference

AmsConfiguration Class [▶ 1543]

TwinCAT.Ams Namespace [▶ 1543]

6.9.2 AmsServerErrorCode Enumeration

Ams Server Error Codes.

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum AmsServerErrorCode
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	ConnectPortFailed	1	Connecting port failed.
	DisconnectPortFailed	2	Disconnecting port failed.
	ReceiveQueueOverflow	3	ReceiveQueue overflow
	ReceiveNotificationQueueOverflow	4	Receive Notification Queue overflow.

Reference

TwinCAT.Ams Namespace [▶ 1543]

6.9.3 AmsServerException Class

Ams Server Exception class Implements the Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.Ams.AmsServerException

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class AmsServerException : Exception
```

The AmsServerException type exposes the following members.

Constructors

	Name	Description
	AmsServerException [▶ 1550]	Initializes a new instance of the AmsServerException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.Ams Namespace [▶ 1543]

System.Exception

6.9.3.1 AmsServerException Constructor

Initializes a new instance of the AmsServerException [▶ 1548] class.

Namespace: TwinCAT.Ams [▶ 1543]
Assembly: TwinCAT.Ads.Server (in TwinCAT.Ads.Server.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected AmsServerException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Exceptions

Exception	Condition
ArgumentNullException	serializationInfo

Reference

AmsServerException Class [▶ 1548]

TwinCAT.Ams Namespace [▶ 1543]

6.9.3.2 AmsServerException Properties

The AmsServerException [▶ 1548] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

AmsServerException Class [▶ 1548]

TwinCAT.Ams Namespace [▶ 1543]

6.9.3.3 AmsServerException Methods

The AmsServerException [▶ 1548] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

AmsServerException Class [▶ 1548]

TwinCAT.Ams Namespace [▶ 1543]

6.9.3.4 AmsServerException Events

The AmsServerException [▶ 1548] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

AmsServerException Class [▶ 1548]

TwinCAT.Ams Namespace [▶ 1543]

6.10 TwinCAT.PlcOpen Namespace

Classes

	Class	Description
	DATE [▶ 1552]	PlcOpen DATE class (32-bit)
	DateBase [▶ 1559]	PlcOpen Date base class (32-Bit)
	DT [▶ 1570]	PlcOpen DT (DATE_AND_TIME) datatype.
	LTIME [▶ 1581]	PlcOpen LTIME class
	LTimeBase [▶ 1589]	Time base class
	TIME [▶ 1598]	PlcOpen TIME class
	TimeBase [▶ 1605]	Base class for PlcOpen Time types.
	TOD [▶ 1615]	PLCOpen TimeOfDay class (32-Bit)

Interfaces

	Interface	Description
	IPlcOpenTimeBase [▶ 1577]	Interface IPlcOpenType
	IPlcOpenTimeBase.T1, T2. [▶ 1579]	Interface IPlcOpenType

6.10.1 DATE Class

PlcOpen DATE class (32-bit)

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.DateBase [▶ 1559]
    TwinCAT.PlcOpen.DATE

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DATE : DateBase
```

The DATE type exposes the following members.

Constructors

	Name	Description
	DATE. [▶ 1554]	Initializes a new instance of the DATE class.
	DATE(DateTimeOffset) [▶ 1554]	Initializes a new instance of the DATE class.
	DATE(Int64) [▶ 1555]	Initializes a new instance of the DATE class.
	DATE(UInt32) [▶ 1555]	Initializes a new instance of the DATE class.
	DATE(Int32, Int32, Int32) [▶ 1556]	Initializes a new instance of the DATE class.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value. (Inherited from DateBase [▶ 1559].)
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase [▶ 1559]. (Inherited from DateBase [▶ 1559].)

Methods

	Name	Description
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Inherited from DateBase [▶ 1559].)
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Inherited from DateBase [▶ 1559].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	Parse [▶ 1557]	Parses the specified string to a DATE object.
	ToString [▶ 1558]	Returns a String that represents this instance. (Overrides Object.ToString..)
S	TryParse [▶ 1558]	Tries to parse the specified string to a DATE object.

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1 DATE Constructor

Overload List

	Name	Description
	DATE. [▶ 1554]	Initializes a new instance of the DATE [▶ 1552] class.
	DATE(DateTimeOffset) [▶ 1554]	Initializes a new instance of the DATE [▶ 1552] class.
	DATE(Int64) [▶ 1555]	Initializes a new instance of the DATE [▶ 1552] class.
	DATE(UInt32) [▶ 1555]	Initializes a new instance of the DATE [▶ 1552] class.
	DATE(Int32, Int32, Int32) [▶ 1556]	Initializes a new instance of the DATE [▶ 1552] class.

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1.1 DATE Constructor

Initializes a new instance of the DATE [▶ 1552] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DATE()
```

Reference

DATE Class [▶ 1552]

DATE Overload [▶ 1554]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1.2 DATE Constructor (DateTimeOffset)

Initializes a new instance of the DATE [▶ 1552] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DATE(
    DateTimeOffset date
)
```

Parameters

date	Type: System.DateTimeOffset The date.

Reference

DATE Class [▶ 1552]

DATE Overload [▶ 1554]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1.3 DATE Constructor (Int64)

Initializes a new instance of the DATE [▶ 1552] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DATE(
    long dateValue
)
```

Parameters

dateValue	Type: System.Int64 The date value in PlcOpen Ticks.

Reference

DATE Class [▶ 1552]

DATE Overload [▶ 1554]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1.4 DATE Constructor (UInt32)

Initializes a new instance of the DATE [▶ 1552] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DATE(
    uint dateValue
)
```

Parameters

dateValue	Type: System.UInt32 The date value in PlcOpen Ticks.

Reference

DATE Class [▶ 1552]

DATE Overload [▶ 1554]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.1.5 DATE Constructor (Int32, Int32, Int32)

Initializes a new instance of the DATE [▶ 1552] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DATE(
    int year,
    int month,
    int day
)
```

Parameters

year	Type: System.Int32 The year.
month	Type: System.Int32 The month.
day	Type: System.Int32 The day.

Reference

DATE Class [▶ 1552]

DATE Overload [▶ 1554]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.2 DATE Properties

The DATE [▶ 1552] type exposes the following members.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value. (Inherited from DateBase [▶ 1559].)
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase [▶ 1559]. (Inherited from DateBase [▶ 1559].)

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.3 DATE Methods

The DATE [▶ 1552] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Inherited from DateBase [▶ 1559].)
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Inherited from DateBase [▶ 1559].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Parse [▶ 1557]	Parses the specified string to a DATE [▶ 1552] object.
	ToString [▶ 1558]	Returns a String that represents this instance. (Overrides Object.ToString..)
	TryParse [▶ 1558]	Tries to parse the specified string to a DATE [▶ 1552] object.

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.3.1 DATE.Parse Method

Parses the specified string to a DATE [▶ 1552] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DATE Parse(
    string s
)
```

Parameters

s	Type: System.String The s.

Return Value

Type: DATE [▶ 1552]
DATE.

Exceptions

Exception	Condition
FormatException	Cannot parse DATE object!

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.3.2 DATE.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.1.3.3 DATE.TryParse Method

Tries to parse the specified string to a DATE [▶ 1552] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string s,
    out DATE date
)
```

Parameters

s	Type: System.String The s.
date	Type: TwinCAT.PlcOpen.DATE [▶ 1552]. The date.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

DATE Class [▶ 1552]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2 DateBase Class

PlcOpen Date base class (32-Bit)

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.DateBase
    TwinCAT.PlcOpen.DATE [▶ 1552]
    TwinCAT.PlcOpen.DT [▶ 1570]

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class DateBase : IPlcOpenTimeBase<DateTimeOffset, uint>,
    IPlcOpenTimeBase
```

The DateBase type exposes the following members.

Constructors

	Name	Description
	DateBase. [▶ 1561]	Initializes a new instance of the DateBase class.
	DateBase(DateTimeOffset) [▶ 1561]	Initializes a new instance of the DateBase class.

	Name	Description
	DateBase(Int64) [▶ 1562]	Initializes a new instance of the DateBase class.
	DateBase(UInt32) [▶ 1562]	Initializes a new instance of the DateBase class.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value.
S	MarshalSize [▶ 1564]	Gets the marshal size in bytes.
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase.

Methods

	Name	Description
S	DateToValue [▶ 1565]	Converts the specified DateTime value to PlcOpen Ticks.
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ParseToTicks [▶ 1567]	Parses the specified PlcOpen Date string to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
S	ValueToDate(Int64) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object
S	ValueToDate(UInt32) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object

Fields

	Name	Description
	plcTimeSeconds [▶ 1569]	PlcTime Seconds (Ticks)

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.1 DateBase Constructor

Overload List

	Name	Description
	DateBase. [▶ 1561]	Initializes a new instance of the DateBase [▶ 1559] class.
	DateBase(DateTimeOffset) [▶ 1561]	Initializes a new instance of the DateBase [▶ 1559] class.
	DateBase(Int64) [▶ 1562]	Initializes a new instance of the DateBase [▶ 1559] class.
	DateBase(UInt32) [▶ 1562]	Initializes a new instance of the DateBase [▶ 1559] class.

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.1.1 DateBase Constructor

Initializes a new instance of the DateBase [▶ 1559] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DateBase()
```

Reference

DateBase Class [▶ 1559]

DateBase Overload [▶ 1561]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.1.2 DateBase Constructor (DateTimeOffset)

Initializes a new instance of the DateBase [▶ 1559] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DateBase(
    DateTimeOffset date
)
```

Parameters

date	Type: System.DateTimeOffset The date.

Reference

DateBase Class [▶ 1559]

DateBase Overload [▶ 1561]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.1.3 DateBase Constructor (Int64)

Initializes a new instance of the DateBase [▶ 1559] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DateBase(
    long dateValue
)
```

Parameters

dateValue	Type: System.Int64 The date value in PlcOpen Ticks.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	

Reference

DateBase Class [▶ 1559]

DateBase Overload [▶ 1561]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.1.4 DateBase Constructor (UInt32)

Initializes a new instance of the DateBase [▶ 1559] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DateBase(
    uint dateValue
)
```

Parameters

dateValue	Type: System.UInt32 The date value in PlcOpen Ticks.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	

Reference

DateBase Class [▶ 1559]

DateBase Overload [▶ 1561]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.2 DateBase Properties

The DateBase [▶ 1559] type exposes the following members.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value.
S	MarshalSize [▶ 1564]	Gets the marshal size in bytes.
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase [▶ 1559].

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.2.1 DateBase.Date Property

Gets or sthe date value.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset Date { get; }
```

Property Value

Type: DateTimeOffset
The date.

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.2.2 DateBase.MarshalSize Property

Gets the marshal size in bytes.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static int MarshalSize { get; }
```

Property Value

Type: Int32
Marshal size in bytes.

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.2.3 DateBase.Ticks Property

Returns the number of ticks that represent the value of this DateBase [▶ 1559].

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Ticks { get; }
```

Property Value

Type: UInt32
The ticks.

Implements

IPlcOpenTimeBase.T1, T2..Ticks [▶ 1580]

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3 DateBase Methods

The DateBase [▶ 1559] type exposes the following members.

Methods

	Name	Description
	DateToValue [▶ 1565]	Converts the specified DateTime value to PlcOpen Ticks.
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ParseToTicks [▶ 1567]	Parses the specified PlcOpen Date string to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	ValueToDate(Int64) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object
	ValueToDate(UInt32) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3.1 DateBase.DateToValue Method

Converts the specified DateTime value to PlcOpen Ticks.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static long DateToValue(
    DateTimeOffset date
)
```

Parameters

date	Type: System.DateTimeOffset The date.

Return Value

Type: Int64

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3.2 DateBase.Equals Method

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3.3 DateBase.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3.4 DateBase.ParseToTicks Method

Parses the specified PlcOpen Date string to PlcOpen ticks.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected abstract long ParseToTicks(
    string s
)
```

Parameters

s	Type: System.String The s.

Return Value

Type: Int64

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.3.5 DateBase.ValueToDate Method

Overload List

	Name	Description
	ValueToDate(Int64) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object
	ValueToDate(UInt32) [▶ 1568]	Converts the specified PlcOpen dateValue in ticks to a DateTime Object

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

DateBase.ValueToDate Method (Int64)

Converts the specified PlcOpen dateValue in ticks to a DateTime Object

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DateTimeOffset ValueToDate(
    long dateValue
)
```

Parameters

dateValue	Type: System.Int64 The date value.

Return Value

Type: DateTimeOffset

Reference

DateBase Class [▶ 1559]

ValueToDate Overload [▶ 1568]

TwinCAT.PlcOpen Namespace [▶ 1552]

DateBase.ValueToDate Method (UInt32)

Converts the specified PlcOpen dateValue in ticks to a DateTime Object

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DateTimeOffset ValueToDate(
    uint dateValue
)
```

Parameters

dateValue	Type: System.UInt32 The date value.

Return Value

Type: DateTimeOffset

Reference

DateBase Class [▶ 1559]

ValueToDate Overload [▶ 1568]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.4 DateBase Fields

The DateBase [▶ 1559] type exposes the following members.

Fields

	Name	Description
	plcTimeSeconds [▶ 1569]	PlcTime Seconds (Ticks)

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.2.4.1 DateBase.plcTimeSeconds Field

PlcTime Seconds (Ticks)

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected uint plcTimeSeconds
```

Field Value

Type: UInt32

Remarks

Seconds from 1/1/1970 in Local Time Zone (different from UnixTimeSeconds which is UTC!). The is exactly the same value and layout, like it is stored in PlcControl for the DATE and DT type!.

Reference

DateBase Class [▶ 1559]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3 DT Class

PlcOpen DT (DATE_AND_TIME) datatype.

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.DateBase [▶ 1559]
    TwinCAT.PlcOpen.DT
Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DT : DateBase
```

The DT type exposes the following members.

Constructors

	Name	Description
	DT. [▶ 1571]	Initializes a new instance of the DT class.
	DT(DateTimeOffset) [▶ 1572]	Initializes a new instance of the DT class.
	DT(Int64) [▶ 1572]	Initializes a new instance of the DT class.
	DT(UInt32) [▶ 1573]	Initializes a new instance of the DT class.
	DT(Int32, Int32, Int32, Int32, Int32, Int32) [▶ 1573]	Initializes a new instance of the DT class.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value. (Inherited from DateBase [▶ 1559].)
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase [▶ 1559]. (Inherited from DateBase [▶ 1559].)

Methods

	Name	Description
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Inherited from DateBase [▶ 1559].)
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Inherited from DateBase [▶ 1559].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	Parse [▶ 1575]	Parses the specified string to the DT object.
	ToString [▶ 1576]	Returns a String that represents this instance. (Overrides Object.ToString..)
S	TryParse [▶ 1576]	Tries to parse the specified string to a DT object.

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1 DT Constructor

Overload List

	Name	Description
	DT. [▶ 1571]	Initializes a new instance of the DT [▶ 1570] class.
	DT(DateTimeOffset) [▶ 1572]	Initializes a new instance of the DT [▶ 1570] class.
	DT(Int64) [▶ 1572]	Initializes a new instance of the DT [▶ 1570] class.
	DT(UInt32) [▶ 1573]	Initializes a new instance of the DT [▶ 1570] class.
	DT(Int32, Int32, Int32, Int32, Int32, Int32) [▶ 1573]	Initializes a new instance of the DT [▶ 1570] class.

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1.1 DT Constructor

Initializes a new instance of the DT [▶ 1570] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DT()
```

Reference

DT Class [▶ 1570]

DT Overload [▶ 1571]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1.2 DT Constructor (DateTimeOffset)

Initializes a new instance of the DT [▶ 1570] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DT(
    DateTimeOffset date
)
```

Parameters

date	Type: System.DateTimeOffset The date.

Reference

DT Class [▶ 1570]

DT Overload [▶ 1571]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1.3 DT Constructor (Int64)

Initializes a new instance of the DT [▶ 1570] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DT(
    long dateValue
)
```

Parameters

dateValue	Type: System.Int64 The date value in PlcOpen Ticks.

Reference

DT Class [▶ 1570]

DT Overload [▶ 1571]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1.4 DT Constructor (UInt32)

Initializes a new instance of the DT [▶ 1570] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DT(
    uint dateValue
)
```

Parameters

dateValue	Type: System.UInt32 The date value in PlcOpen Ticks.

Reference

DT Class [▶ 1570]

DT Overload [▶ 1571]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.1.5 DT Constructor (Int32, Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the DT [▶ 1570] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DT(
    int year,
    int month,
    int day,
    int hour,
```

```
    int minute,
    int second
)
```

Parameters

year	Type: System.Int32 The year.
month	Type: System.Int32 The month.
day	Type: System.Int32 The day.
hour	Type: System.Int32 The hour.
minute	Type: System.Int32 The minute.
second	Type: System.Int32 The second.

Reference

DT Class [▶ 1570]

DT Overload [▶ 1571]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.2 DT Properties

The DT [▶ 1570] type exposes the following members.

Properties

	Name	Description
	Date [▶ 1563]	Gets or sthe date value. (Inherited from DateBase [▶ 1559].)
	Ticks [▶ 1564]	Returns the number of ticks that represent the value of this DateBase [▶ 1559]. (Inherited from DateBase [▶ 1559].)

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.3 DT Methods

The DT [▶ 1570] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1566]	Determines whether the specified Object is equal to this instance. (Inherited from DateBase [▶ 1559].)

	Name	Description
	GetHashCode [▶ 1566]	Gets the HashCode of the Address (Inherited from DateBase [▶ 1559].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	Parse [▶ 1575]	Parses the specified string to the DT [▶ 1570] object.
	ToString [▶ 1576]	Returns a String that represents this instance. (Overrides Object.ToString..)
S	TryParse [▶ 1576]	Tries to parse the specified string to a DT [▶ 1570] object.

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.3.1 DT.Parse Method

Parses the specified string to the DT [▶ 1570] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DT Parse(
    string s
)
```

Parameters

s	Type: System.String The s.

Return Value

Type: DT [▶ 1570]
DT.

Exceptions

Exception	Condition
FormatException	Cannot parse DT object!

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.3.2 DT.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.3.3.3 DT.TryParse Method

Tries to parse the specified string to a DT [▶ 1570] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string s,
    out DT dt
)
```

Parameters

s	Type: System.String The s.
dt	Type: TwinCAT.PlcOpen.DT [▶ 1570]. The dt.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

DT Class [▶ 1570]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.4 IPlcOpenTimeBase Interface

Interface IPlcOpenType

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IPlcOpenTimeBase
```

The IPlcOpenTimeBase type exposes the following members.

Properties

	Name	Description
	ManagedValueType [▶ 1577]	Gets the type of the underlying human readable type (DateTime or Timespan)
	TicksValueType [▶ 1578]	Gets the type of the underlying ticks resolution (uint32 or uint64)
	UntypedValue [▶ 1578]	Returns the 'Value' as object type.

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.4.1 IPlcOpenTimeBase Properties

The IPlcOpenTimeBase [▶ 1577] type exposes the following members.

Properties

	Name	Description
	ManagedValueType [▶ 1577]	Gets the type of the underlying human readable type (DateTime or Timespan)
	TicksValueType [▶ 1578]	Gets the type of the underlying ticks resolution (uint32 or uint64)
	UntypedValue [▶ 1578]	Returns the 'Value' as object type.

Reference

IPlcOpenTimeBase Interface [▶ 1577]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.4.1.1 IPlcOpenTimeBase.ManagedValueType Property

Gets the type of the underlying human readable type (DateTime or Timespan)

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Type ManagedValueType { get; }
```

Property Value

Type: Type
The type of the managed value.

Reference

IPlcOpenTimeBase Interface [▶ 1577]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.4.1.2 IPlcOpenTimeBase.TicksValueType Property

Gets the type of the underlying ticks resolution (uint32 or uint64)

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Type TicksValueType { get; }
```

Property Value

Type: Type
The type of the ticks value.

Reference

IPlcOpenTimeBase Interface [▶ 1577]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.4.1.3 IPlcOpenTimeBase.UntypedValue Property

Returns the 'Value' as object type.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object UntypedValue { get; }
```

Property Value

Type: Object
The untyped value.

Reference

IPlcOpenTimeBase Interface [▶ 1577]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.5 IPlcOpenTimeBase.T1, T2. Interface

Interface IPlcOpenType

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IPlcOpenTimeBase<T1, T2> : IPlcOpenTimeBase
```

Type Parameters

T1	The type of the t1.
T2	The type of the t2.

The IPlcOpenTimeBase.T1, T2. type exposes the following members.

Properties

	Name	Description
	ManagedValueType [▶ 1577]	Gets the type of the underlying human readable type (DateTime or Timespan) (Inherited from IPlcOpenTimeBase [▶ 1577].)
	Ticks [▶ 1580]	Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. (uint32 or uint64).
	TicksValueType [▶ 1578]	Gets the type of the underlying ticks resolution (uint32 or uint64) (Inherited from IPlcOpenTimeBase [▶ 1577].)
	UntypedValue [▶ 1578]	Returns the 'Value' as object type. (Inherited from IPlcOpenTimeBase [▶ 1577].)
	Value [▶ 1580]	Returns the value of this IPlcOpenTimeBase.T1, T2. as Managed base type (DateTime or Timespan)

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

TwinCAT.PlcOpen.IPlcOpenTimeBase [▶ 1577]

6.10.5.1 IPlcOpenTimeBase.T1, T2. Properties

The IPlcOpenTimeBase.T1, T2. [▶ 1579] generic type exposes the following members.

Properties

	Name	Description
	ManagedValueType [▶ 1577]	Gets the type of the underlying human readable type (DateTime or Timespan) (Inherited from IPlcOpenTimeBase [▶ 1577].)
	Ticks [▶ 1580]	Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. [▶ 1579] (uint32 or uint64).
	TicksValueType [▶ 1578]	Gets the type of the underlying ticks resolution (uint32 or uint64) (Inherited from IPlcOpenTimeBase [▶ 1577].)
	UntypedValue [▶ 1578]	Returns the 'Value' as object type. (Inherited from IPlcOpenTimeBase [▶ 1577].)
	Value [▶ 1580]	Returns the value of this IPlcOpenTimeBase.T1, T2. [▶ 1579] as Managed base type (DateTime or Timespan)

Reference

IPlcOpenTimeBase.T1, T2. Interface [▶ 1579]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.5.1.1 IPlcOpenTimeBase.T1, T2..Ticks Property

Returns the number of ticks that represent the value of this IPlcOpenTimeBase.T1, T2. [▶ 1579] (uint32 or uint64).

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T2 Ticks { get; }
```

Property Value

Type: T2 [▶ 1579]
The ticks.

Reference

IPlcOpenTimeBase.T1, T2. Interface [▶ 1579]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.5.1.2 IPlcOpenTimeBase.T1, T2..Value Property

Returns the value of this IPlcOpenTimeBase.T1, T2. [▶ 1579] as Managed base type (DateTime or Timespan)

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T1 Value { get; }
```

Property Value

Type: T1 [▶ 1579]
The value.

Reference

IPlcOpenTimeBase.T1, T2. Interface [▶ 1579]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6 LTIME Class

PlcOpen LTIME class

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.LTimeBase [▶ 1589]
    TwinCAT.PlcOpen.LTIME
Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class LTIME : LTimeBase
```

The LTIME type exposes the following members.

Constructors

	Name	Description
	LTIME. [▶ 1583]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(Int64) [▶ 1583]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(TimeSpan) [▶ 1584]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(UInt64) [▶ 1584]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(Int32, Int32, Int32) [▶ 1585]	Initializes a new instance of the LTIME class.
	LTIME(Int32, Int32, Int32, Int32) [▶ 1585]	Initializes a new instance of the LTIME class.
	LTIME(Int32, Int32, Int32, Int32, Int32, Int32, Int32) [▶ 1586]	Initializes a new instance of the LTIME class.

Properties

	Name	Description
	Ticks [▶ 1593]	Returns the number of ticks that represent the value of this LTimeBase [▶ 1589]. (Inherited from LTimeBase [▶ 1589].)
	Time [▶ 1593]	Gets or the time value. (Inherited from LTimeBase [▶ 1589].)

Methods

	Name	Description
	Equals [▶ 1594]	Determines whether the specified Object is equal to this instance. (Inherited from LTimeBase [▶ 1589].)
	GetHashCode [▶ 1595]	Gets the HashCode of the Address (Inherited from LTimeBase [▶ 1589].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Parse [▶ 1587]	Parses the specified string to a LTIME object.
	ToString [▶ 1588]	Returns a String that represents this instance. (Overrides Object.ToString..)
	TryParse [▶ 1588]	Tries to parse the string to a LTIME object.

Remarks

This class has an internal requesentation of an UINT64 (8 Bytes)

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1 LTIME Constructor

Overload List

	Name	Description
	LTIME. [▶ 1583]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(Int64) [▶ 1583]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(TimeSpan) [▶ 1584]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(UInt64) [▶ 1584]	Initializes a new instance of the TIME [▶ 1598] class.
	LTIME(Int32, Int32, Int32) [▶ 1585]	Initializes a new instance of the LTIME [▶ 1581] class.
	LTIME(Int32, Int32, Int32, Int32) [▶ 1585]	Initializes a new instance of the LTIME [▶ 1581] class.
	LTIME(Int32, Int32, Int32, Int32, Int32, Int32) [▶ 1586]	Initializes a new instance of the LTIME [▶ 1581] class.

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.1 LTIME Constructor

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME()
```

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.2 LTIME Constructor (Int64)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    long timeValue
)
```

Parameters

timeValue	Type: System.Int64 The time value.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.3 LTIME Constructor (TimeSpan)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    TimeSpan time
)
```

Parameters

time	Type: System.TimeSpan The time.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.4 LTIME Constructor (UInt64)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    ulong timeValue
)
```

Parameters

timeValue	Type: System.UInt64 The time value.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.5 LTIME Constructor (Int32, Int32, Int32)

Initializes a new instance of the LTIME [▶ 1581] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    int seconds,
    int milliseconds,
    int microseconds
)
```

Parameters

seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.
microseconds	Type: System.Int32 The microseconds.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.6 LTIME Constructor (Int32, Int32, Int32, Int32)

Initializes a new instance of the LTIME [▶ 1581] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    int seconds,
    int milliseconds,
    int microseconds,
    int nanoseconds
)
```

Parameters

seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.

microseconds	Type: System.Int32 The microseconds.
nanoseconds	Type: System.Int32 The nanoseconds.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.1.7 LTIME Constructor (Int32, Int32, Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the LTIME [▶ 1581] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public LTIME(
    int days,
    int hours,
    int minutes,
    int seconds,
    int milliseconds,
    int microseconds,
    int nanoseconds
)
```

Parameters

days	Type: System.Int32 The days.
hours	Type: System.Int32 The hours.
minutes	Type: System.Int32 The minutes.
seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.
microseconds	Type: System.Int32 The microseconds.
nanoseconds	Type: System.Int32 The nanoseconds.

Reference

LTIME Class [▶ 1581]

LTIME Overload [▶ 1582]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.2 LTIME Properties

The LTIME [▶ 1581] type exposes the following members.

Properties

	Name	Description
	Ticks [▶ 1593]	Returns the number of ticks that represent the value of this LTimeBase [▶ 1589]. (Inherited from LTimeBase [▶ 1589].)
	Time [▶ 1593]	Gets or the time value. (Inherited from LTimeBase [▶ 1589].)

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.3 LTIME Methods

The LTIME [▶ 1581] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1594]	Determines whether the specified Object is equal to this instance. (Inherited from LTimeBase [▶ 1589].)
	GetHashCode [▶ 1595]	Gets the HashCode of the Address (Inherited from LTimeBase [▶ 1589].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Parse [▶ 1587]	Parses the specified string to a LTIME [▶ 1581] object.
	ToString [▶ 1588]	Returns a String that represents this instance. (Overrides Object.ToString..)
	TryParse [▶ 1588]	Tries to parse the string to a LTIME [▶ 1581] object.

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.3.1 LTIME.Parse Method

Parses the specified string to a LTIME [▶ 1581] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static LTIME Parse(
    string str
)
```

Parameters

str	Type: System.String The string.

Return Value

Type: LTIME [▶ 1581]
LTIME.

Exceptions

Exception	Condition
FormatException	Cannot create TIME DataType!

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.3.2 LTIME.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.6.3.3 LTIME.TryParse Method

Tries to parse the string to a LTIME [▶ 1581] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string str,
    out LTIME ret
)
```

Parameters

str	Type: System.String The string.
ret	Type: TwinCAT.PlcOpen.LTIME [▶ 1581]. The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

LTIME Class [▶ 1581]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7 LTimeBase Class

Time base class

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.LTimeBase
    TwinCAT.PlcOpen.LTIME [▶ 1581]
Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class LTimeBase : IPlcOpenTimeBase<TimeSpan, ulong>,
    IPlcOpenTimeBase
```

The LTimeBase type exposes the following members.

Constructors

	Name	Description
	LTimeBase. [▶ 1591]	Initializes a new instance of the TimeBase [▶ 1605] class.

	Name	Description
	LTimeBase(UInt64) [▶ 1591]	Initializes a new instance of the TimeBase [▶ 1605] class.

Properties

	Name	Description
S	MarshalSize [▶ 1592]	Gets the marshal size in bytes.
	Ticks [▶ 1593]	Returns the number of ticks that represent the value of this LTimeBase.
	Time [▶ 1593]	Gets or the time value.

Methods

	Name	Description
	Equals [▶ 1594]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1595]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	TimeToValue [▶ 1595]	Converts the Timespan to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
S	ValueToTime(Int64) [▶ 1596]	Converts the timeValue (PlcOpen ticks) to TimeSpan
S	ValueToTime(UInt64) [▶ 1597]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Fields

	Name	Description
	internalTimeValue [▶ 1597]	The internal time value

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.1 LTimeBase Constructor

Overload List

	Name	Description
	LTimeBase. [▶ 1591]	Initializes a new instance of the TimeBase [▶ 1605] class.
	LTimeBase(UInt64) [▶ 1591]	Initializes a new instance of the TimeBase [▶ 1605] class.

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.1.1 LTimeBase Constructor

Initializes a new instance of the TimeBase [▶ 1605] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected LTimeBase()
```

Reference

LTimeBase Class [▶ 1589]

LTimeBase Overload [▶ 1591]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.1.2 LTimeBase Constructor (UInt64)

Initializes a new instance of the TimeBase [▶ 1605] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected LTimeBase(
    ulong timeValue
)
```

Parameters

timeValue	Type: System.UInt64 The time value.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	

Reference

LTimeBase Class [▶ 1589]

LTimeBase Overload [▶ 1591]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.2 LTimeBase Properties

The LTimeBase [▶ 1589] type exposes the following members.

Properties

	Name	Description
	MarshalSize [▶ 1592]	Gets the marshal size in bytes.
	Ticks [▶ 1593]	Returns the number of ticks that represent the value of this LTimeBase [▶ 1589].
	Time [▶ 1593]	Gets or the time value.

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.2.1 LTimeBase.MarshalSize Property

Gets the marshal size in bytes.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static int MarshalSize { get; }
```

Property Value

Type: Int32
Marshal size in bytes.

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.2.2 LTimeBase.Ticks Property

Returns the number of ticks that represent the value of this LTimeBase [▶ 1589].

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ulong Ticks { get; }
```

Property Value

Type: UInt64
The ticks.

Implements

IPlcOpenTimeBase.T1, T2..Ticks [▶ 1580]

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.2.3 LTimeBase.Time Property

Gets or the time value.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TimeSpan Time { get; }
```

Property Value

Type: TimeSpan
The time.

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.3 LTimeBase Methods

The LTimeBase [▶ 1589] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1594]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1595]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	TimeToValue [▶ 1595]	Converts the Timespan to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
S	ValueToTime(Int64) [▶ 1596]	Converts the timeValue (PlcOpen ticks) to TimeSpan
S	ValueToTime(UInt64) [▶ 1597]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.3.1 LTimeBase.Equals Method

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.3.2 LTimeBase.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.3.3 LTimeBase.TimeToValue Method

Converts the Timespan to PlcOpen ticks.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ulong TimeToValue(
    TimeSpan time
)
```

Parameters

time	Type: System.TimeSpan The time.

Return Value

Type: UInt64

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.3.4 LTimeBase.ValueToTime Method

Overload List

	Name	Description
	ValueToTime(Int64) [▶ 1596]	Converts the timeValue (PlcOpen ticks) to TimeSpan
	ValueToTime(UInt64) [▶ 1597]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

LTimeBase.ValueToTime Method (Int64)

Converts the timeValue (PlcOpen ticks) to TimeSpan

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan ValueToTime(
    long nanoseconds
)
```

Parameters

nanoseconds	Type: System.Int64 The time value.

Return Value

Type: TimeSpan

Reference

LTimeBase Class [▶ 1589]

ValueToTime Overload [▶ 1596]

TwinCAT.PlcOpen Namespace [▶ 1552]

LTimeBase.ValueToTime Method (UInt64)

Converts the timeValue (PlcOpen ticks) to TimeSpan

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan ValueToTime(
    ulong nanoseconds
)
```

Parameters

nanoseconds	Type: System.UInt64 The time value.

Return Value

Type: TimeSpan

Reference

LTimeBase Class [▶ 1589]

ValueToTime Overload [▶ 1596]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.4 LTimeBase Fields

The LTimeBase [▶ 1589] type exposes the following members.

Fields

	Name	Description
	internalTimeValue [▶ 1597]	The internal time value

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.7.4.1 LTimeBase.internalTimeValue Field

The internal time value

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected ulong internalTimeValue
```

Field Value

Type: UInt64

Reference

LTimeBase Class [▶ 1589]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8 TIME Class

PlcOpen TIME class

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.TimeBase [▶ 1605]
    TwinCAT.PlcOpen.TIME
Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class TIME : TimeBase
```

The TIME type exposes the following members.

Constructors

	Name	Description
	TIME. [▶ 1600]	Initializes a new instance of the TIME class.
	TIME(Int64) [▶ 1600]	Initializes a new instance of the TIME class.
	TIME(TimeSpan) [▶ 1601]	Initializes a new instance of the TIME class.
	TIME(UInt32) [▶ 1601]	Initializes a new instance of the TIME class.
	TIME(Int32, Int32) [▶ 1602]	Initializes a new instance of the TIME class.
	TIME(Int32, Int32, Int32, Int32, Int32) [▶ 1602]	Initializes a new instance of the TIME class.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value. (Inherited from TimeBase [▶ 1605].)
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64). (Inherited from TimeBase [▶ 1605].)
	Time [▶ 1611]	Gets the time value. (Inherited from TimeBase [▶ 1605].)

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [▶ 1605].)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Inherited from TimeBase [▶ 1605].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	Parse [▶ 1604]	Parses the specified string to a TIME object.
	ToString [▶ 1604]	Returns a String that represents this instance. (Overrides Object.ToString..)
S	TryParse [▶ 1605]	Tries to parse the TIME object from string.

Remarks

This class has an internal representation of an UINT32 (4 bytes).

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1 TIME Constructor

Overload List

	Name	Description
	TIME. [▶ 1600]	Initializes a new instance of the TIME [▶ 1598] class.
	TIME(Int64) [▶ 1600]	Initializes a new instance of the TIME [▶ 1598] class.
	TIME(TimeSpan) [▶ 1601]	Initializes a new instance of the TIME [▶ 1598] class.
	TIME(UInt32) [▶ 1601]	Initializes a new instance of the TIME [▶ 1598] class.
	TIME(Int32, Int32) [▶ 1602]	Initializes a new instance of the TIME [▶ 1598] class.
	TIME(Int32, Int32, Int32, Int32, Int32) [▶ 1602]	Initializes a new instance of the TIME [▶ 1598] class.

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.1 TIME Constructor

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME()
```

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.2 TIME Constructor (Int64)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME(
    long timeValue
)
```

Parameters

timeValue	Type: System.Int64 The time value.

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.3 TIME Constructor (TimeSpan)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME(
    TimeSpan time
)
```

Parameters

time	Type: System.TimeSpan The time.

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.4 TIME Constructor (UInt32)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME(
    uint timeValue
)
```

Parameters

timeValue	Type: System.UInt32 The time value.

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.5 TIME Constructor (Int32, Int32)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME(
    int seconds,
    int milliseconds
)
```

Parameters

seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.1.6 TIME Constructor (Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the TIME [▶ 1598] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TIME(
    int days,
    int hours,
    int minutes,
    int seconds,
    int milliseconds
)
```

Parameters

days	Type: System.Int32 The days.
hours	Type: System.Int32 The hours.
minutes	Type: System.Int32 The minutes.

seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.

Reference

TIME Class [▶ 1598]

TIME Overload [▶ 1599]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.2 TIME Properties

The TIME [▶ 1598] type exposes the following members.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value. (Inherited from TimeBase [▶ 1605].)
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64). (Inherited from TimeBase [▶ 1605].)
	Time [▶ 1611]	Gets the time value. (Inherited from TimeBase [▶ 1605].)

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.3 TIME Methods

The TIME [▶ 1598] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [▶ 1605].)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Inherited from TimeBase [▶ 1605].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Parse [▶ 1604]	Parses the specified string to a TIME [▶ 1598] object.
	ToString [▶ 1604]	Returns a String that represents this instance. (Overrides Object.ToString..)
	TryParse [▶ 1605]	Tries to parse the TIME [▶ 1598] object from string.

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.3.1 TIME.Parse Method

Parses the specified string to a TIME [▶ 1598] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TIME Parse(
    string str
)
```

Parameters

str	Type: System.String The string.

Return Value

Type: TIME [▶ 1598]
TIME.

Exceptions

Exception	Condition
FormatException	Cannot create TIME DataType!

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.3.2 TIME.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.8.3.3 TIME.TryParse Method

Tries to parse the TIME [▶ 1598] object from string.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string str,
    out TIME ret
)
```

Parameters

str	Type: System.String The string.
ret	Type: TwinCAT.PlcOpen.TIME [▶ 1598]. The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TIME Class [▶ 1598]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9 TimeBase Class

Base class for PlcOpen Time types.

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.TimeBase
    TwinCAT.PlcOpen.TIME [▶ 1598]
    TwinCAT.PlcOpen.TOD [▶ 1615]

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class TimeBase : IPlcOpenTimeBase<TimeSpan, uint>,
    IPlcOpenTimeBase
```

The TimeBase type exposes the following members.

Constructors

	Name	Description
	TimeBase. [▶ 1607]	Initializes a new instance of the TimeBase class.
	TimeBase(Int64) [▶ 1608]	Initializes a new instance of the TimeBase class.
	TimeBase(UInt32) [▶ 1608]	Initializes a new instance of the TimeBase class.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value.
S	MarshalSize [▶ 1610]	Gets the marshal size in bytes.
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase (uint32 or uint64).
	Time [▶ 1611]	Gets the time value.

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	TimeToValue [▶ 1613]	Converts the Timespan to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

	Name	Description
	ValueToTime(Int64) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan
	ValueToTime(UInt32) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Fields

	Name	Description
	internalTimeValue [▶ 1615]	The internal time value

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.1 TimeBase Constructor

Overload List

	Name	Description
	TimeBase. [▶ 1607]	Initializes a new instance of the TimeBase [▶ 1605] class.
	TimeBase(Int64) [▶ 1608]	Initializes a new instance of the TimeBase [▶ 1605] class.
	TimeBase(UInt32) [▶ 1608]	Initializes a new instance of the TimeBase [▶ 1605] class.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.1.1 TimeBase Constructor

Initializes a new instance of the TimeBase [▶ 1605] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected TimeBase()
```

Reference

TimeBase Class [▶ 1605]

TimeBase Overload [▶ 1607]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.1.2 TimeBase Constructor (Int64)

Initializes a new instance of the TimeBase [▶ 1605] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected TimeBase(
    long timeValue
)
```

Parameters

timeValue	Type: System.Int64 The time value.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	

Reference

TimeBase Class [▶ 1605]

TimeBase Overload [▶ 1607]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.1.3 TimeBase Constructor (UInt32)

Initializes a new instance of the TimeBase [▶ 1605] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected TimeBase(
    uint timeValue
)
```

Parameters

timeValue	Type: System.UInt32 The time value.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	

Reference

TimeBase Class [▶ 1605]

TimeBase Overload [▶ 1607]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.2 TimeBase Properties

The TimeBase [▶ 1605] type exposes the following members.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value.
	MarshalSize [▶ 1610]	Gets the marshal size in bytes.
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64).
	Time [▶ 1611]	Gets the time value.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.2.1 TimeBase.InternalTimeValue Property

Gets the internal time value.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint InternalTimeValue { get; }
```

Property Value

Type: UInt32
The internal time value.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.2.2 TimeBase.MarshalSize Property

Gets the marshal size in bytes.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static int MarshalSize { get; }
```

Property Value

Type: Int32
Marshal size in bytes.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.2.3 TimeBase.Ticks Property

Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64).

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public uint Ticks { get; }
```

Property Value

Type: UInt32
The ticks (in 100ns).

Implements

IPlcOpenTimeBase.T1, T2..Ticks [▶ 1580]

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.2.4 TimeBase.Time Property

Gets the time value.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual TimeSpan Time { get; }
```

Property Value

Type: TimeSpan
The time.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.3 TimeBase Methods

The TimeBase [▶ 1605] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	TimeToValue [▶ 1613]	Converts the Timespan to PlcOpen ticks.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
S	ValueToTime(Int64) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan
S	ValueToTime(UInt32) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.3.1 TimeBase.Equals Method

Determines whether the specified Object is equal to this instance.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.3.2 TimeBase.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.3.3 TimeBase.TimeToValue Method

Converts the Timespan to PlcOpen ticks.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static long TimeToValue(
    TimeSpan time
)
```

Parameters

time	Type: System.TimeSpan The time.

Return Value

Type: Int64

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.3.4 TimeBase.ValueToTime Method

Overload List

	Name	Description
	ValueToTime(Int64) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan
	ValueToTime(UInt32) [▶ 1614]	Converts the timeValue (PlcOpen ticks) to TimeSpan

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

TimeBase.ValueToTime Method (Int64)

Converts the timeValue (PlcOpen ticks) to TimeSpan

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan ValueToTime(
    long timeValue
)
```

Parameters

timeValue	Type: System.Int64 The time value.

Return Value

Type: TimeSpan

Reference

TimeBase Class [▶ 1605]

ValueToTime Overload [▶ 1613]

TwinCAT.PlcOpen Namespace [▶ 1552]

TimeBase.ValueToTime Method (UInt32)

Converts the timeValue (PlcOpen ticks) to TimeSpan

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TimeSpan ValueToTime(
    uint timeValue
)
```

Parameters

timeValue	Type: System.UInt32 The time value.

Return Value

Type: TimeSpan

Reference

TimeBase Class [▶ 1605]

ValueToTime Overload [▶ 1613]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.4 TimeBase Fields

The TimeBase [▶ 1605] type exposes the following members.

Fields

	Name	Description
	internalTimeValue [▶ 1615]	The internal time value

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.9.4.1 TimeBase.internalTimeValue Field

The internal time value

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected uint internalTimeValue
```

Field Value

Type: UInt32

Reference

TimeBase Class [▶ 1605]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10 TOD Class

PLCOpen TimeOfDay class (32-Bit)

Inheritance Hierarchy

System.Object
  TwinCAT.PlcOpen.TimeBase [▶ 1605]
    TwinCAT.PlcOpen.TOD

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class TOD : TimeBase
```

The TOD type exposes the following members.

Constructors

	Name	Description
	TOD. [▶ 1617]	Initializes a new instance of the TOD class.
	TOD(Int64) [▶ 1617]	Initializes a new instance of the TOD class.
	TOD(TimeSpan) [▶ 1618]	Initializes a new instance of the TOD class.
	TOD(UInt32) [▶ 1618]	Initializes a new instance of the TOD class.
	TOD(Int32, Int32, Int32, Int32, Int32) [▶ 1619]	Initializes a new instance of the TOD class.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value. (Inherited from TimeBase [▶ 1605].)
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64). (Inherited from TimeBase [▶ 1605].)
	Time [▶ 1611]	Gets the time value. (Inherited from TimeBase [▶ 1605].)

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [▶ 1605].)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Inherited from TimeBase [▶ 1605].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
S	Parse [▶ 1620]	Parses the specified string to a TOD object.
	ToString [▶ 1621]	Returns a string that represents the current object. (Overrides Object.ToString..)
S	TryParse [▶ 1621]	Tries to parse the string to a TOD object.

Reference

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1 TOD Constructor

Overload List

	Name	Description
	TOD. [▶ 1617]	Initializes a new instance of the TOD [▶ 1615] class.
	TOD(Int64) [▶ 1617]	Initializes a new instance of the TOD [▶ 1615] class.
	TOD(TimeSpan) [▶ 1618]	Initializes a new instance of the TOD [▶ 1615] class.
	TOD(UInt32) [▶ 1618]	Initializes a new instance of the TOD [▶ 1615] class.
	TOD(Int32, Int32, Int32, Int32, Int32) [▶ 1619]	Initializes a new instance of the TOD [▶ 1615] class.

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1.1 TOD Constructor

Initializes a new instance of the TOD [▶ 1615] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TOD()
```

Reference

TOD Class [▶ 1615]

TOD Overload [▶ 1617]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1.2 TOD Constructor (Int64)

Initializes a new instance of the TOD [▶ 1615] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TOD(
    long time
)
```

Parameters

time	Type: System.Int64 The time.

Reference

TOD Class [▶ 1615]

TOD Overload [▶ 1617]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1.3 TOD Constructor (TimeSpan)

Initializes a new instance of the TOD [▶ 1615] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TOD(
    TimeSpan timeSpan
)
```

Parameters

timeSpan	Type: System.TimeSpan The time span.

Reference

TOD Class [▶ 1615]

TOD Overload [▶ 1617]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1.4 TOD Constructor (UInt32)

Initializes a new instance of the TOD [▶ 1615] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TOD(
    uint time
)
```

Parameters

time	Type: System.UInt32 The time.

Reference

TOD Class [▶ 1615]

TOD Overload [▶ 1617]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.1.5 TOD Constructor (Int32, Int32, Int32, Int32, Int32)

Initializes a new instance of the TOD [▶ 1615] class.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TOD(
    int days,
    int hours,
    int minutes,
    int seconds,
    int milliseconds
)
```

Parameters

days	Type: System.Int32 The days.
hours	Type: System.Int32 The hours.
minutes	Type: System.Int32 The minutes.
seconds	Type: System.Int32 The seconds.
milliseconds	Type: System.Int32 The milliseconds.

Reference

TOD Class [▶ 1615]

TOD Overload [▶ 1617]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.2 TOD Properties

The TOD [▶ 1615] type exposes the following members.

Properties

	Name	Description
	InternalTimeValue [▶ 1609]	Gets the internal time value. (Inherited from TimeBase [▶ 1605].)
	Ticks [▶ 1610]	Returns the number of ticks that represent the value of this TimeBase [▶ 1605] (uint32 or uint64). (Inherited from TimeBase [▶ 1605].)
	Time [▶ 1611]	Gets the time value. (Inherited from TimeBase [▶ 1605].)

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.3 TOD Methods

The TOD [▶ 1615] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1612]	Determines whether the specified Object is equal to this instance. (Inherited from TimeBase [▶ 1605].)
	GetHashCode [▶ 1612]	Gets the HashCode of the Address (Inherited from TimeBase [▶ 1605].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	Parse [▶ 1620]	Parses the specified string to a TOD [▶ 1615] object.
	ToString [▶ 1621]	Returns a string that represents the current object. (Overrides Object.ToString..)
	TryParse [▶ 1621]	Tries to parse the string to a TOD [▶ 1615] object.

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.3.1 TOD.Parse Method

Parses the specified string to a TOD [▶ 1615] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TOD Parse(
    string str
)
```

Parameters

str	Type: System.String The string.

Return Value

Type: TOD [▶ 1615]
TOD.

Exceptions

Exception	Condition
FormatException	Cannot parse TOD object!

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.3.2 TOD.ToString Method

Returns a string that represents the current object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A string that represents the current object.

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.10.10.3.3 TOD.TryParse Method

Tries to parse the string to a TOD [▶ 1615] object.

Namespace: TwinCAT.PlcOpen [▶ 1552]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    string str,
    out TOD ret
)
```

Parameters

str	Type: System.String The string.
ret	Type: TwinCAT.PlcOpen.TOD [▶ 1615]. The ret.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TOD Class [▶ 1615]

TwinCAT.PlcOpen Namespace [▶ 1552]

6.11 TwinCAT.TypeSystem Namespace

Namespace for the common (non ADS dependent) type system.

Classes

	Class	Description
	AnySymbolSpecifier [▶ 1629]	Class AnySymbolSpecifier.
	AnyTypeSpecifier [▶ 1633]	Class AnyTypeSpecifier.
	CannotAccessVirtualSymbolException [▶ 1641]	Cannot access virtual Symbol
	CannotResolveDataTypeException [▶ 1644]	Class CannotResolveDataTypeException. Implements the DataTypeException [▶ 1658]
	DataTypeCollection [▶ 1650]	Collection of DataTypes. [▶ 1986]
	DataTypeEventArgs [▶ 1656]	Class DataTypeEventArgs.
	DataTypeException [▶ 1658]	Data Type Exception

	Class	Description
	DataTypeNameEventArgs [▶ 1665]	Class DataTypeNameEventArgs.
	Dimension [▶ 1668]	Represents a single dimension of an IArrayType [▶ 1971]
	DimensionCollection [▶ 1671]	Collection class for Array Dimensions
	DynamicAliasInstance [▶ 1687]	Class DynamicAliasInstance. This class cannot be inherited.
	DynamicArrayInstance [▶ 1698]	Dynamic Array Instance
	DynamicOversamplingArrayInstance [▶ 1712]	Dynamic Array Instance
	DynamicPointerInstance [▶ 1720]	Dynamic Pointer Instance
	DynamicPointerValue [▶ 1728]	Class DynamicPointerValue.
	DynamicReferenceInstance [▶ 1734]	Dynamic Reference Instance
	DynamicReferenceValue [▶ 1747]	Class DynamicReferenceValue.
	DynamicRpcStructInstance [▶ 1752]	Dynamic struct instance with RPC Methods.
	DynamicStructInstance [▶ 1779]	Dynamic struct instance
	DynamicSymbol [▶ 1791]	Dynamic Symbol [▶ 2176] object.
	DynamicSymbolsCollection [▶ 1853]	Dynamic (Expandable) Symbols collection.
	DynamicUnionInstance [▶ 1860]	Dynamic union instance
	DynamicValue [▶ 1869]	Dynamic value (uses RuntimeBinding for ISymbol [▶ 2176] value reading / writing).

	Class	Description
	DynamicVirtualStructInstance [▶ 1894]	Dynamic struct instance
	EnumValue.T. [▶ 1901]	Enum Value
	EnumValueCollection [▶ 1908]	Class EnumValueCollection.
	EnumValueCollection.T. [▶ 1926]	Collection of EnumValues [▶ 1901]
	FieldCollection [▶ 1945]	Collection of IField [▶ 2040] objects.
	InsufficientAccessRightsException [▶ 2076]	Insufficient rights for access
	MarshalException [▶ 2273]	Common Marshalling Exception
	MemberCollection [▶ 2281]	Collection of IMember [▶ 2065] objects.
	RawValueChangedEventArgs [▶ 2289]	Event args for the RawValueChanged [▶ 2254] event.
	ReadOnlyDataTypeCollection [▶ 2291]	ReadOnly Collection of IDataType [▶ 1986] objects.
	ReadOnlyDimensionCollection [▶ 2295]	ReadOnly version of the DimensionCollection [▶ 1671]
	ReadOnlyEnumValueCollection [▶ 2300]	Read only version of the EnumValueCollection.T. [▶ 1926]
	ReadOnlyEnumValueCollection.T. [▶ 2309]	Read only version of the EnumValueCollection.T. [▶ 1926]
	ReadOnlyFieldCollection [▶ 2317]	Read only collection of IField [▶ 2040] objects
	ReadOnlyMemberCollection [▶ 2322]	Read only collection of IMember [▶ 2065] objects
	ReadOnlyMethodParameterCollection [▶ 2327]	Read only RpcMethodParameterCollection [▶ 2384].
	ReadOnlyRpcMethodCollection [▶ 2330]	Read only RpcMethodCollection [▶ 2364]

	Class	Description
	ReadOnlySymbolCollection [▶ 2336]	ReadOnly collection containing ISymbol [▶ 2176] objects.
	ReadOnlyTypeAttributeCollection [▶ 2340]	Read only version of the TypeAttributeCollection [▶ 2418]
	ResultDataTypes [▶ 2347]	Class representing the asynchronous result of reading a IDataTypeCollection [▶ 1993] via ADS. Implements the ResultValue.TValue. [▶ 1029]
	ResultDynamicSymbols [▶ 2350]	Class representing the asynchronous result of reading a dynamic symbol collection via ADS. Implements the ResultValue.TValue. [▶ 1029]
	ResultSymbols [▶ 2353]	Class representing the asynchronous result of reading an symbol collection of type ISymbolCollection.T. [▶ 2185] via ADS. Implements the ResultValue.TValue. [▶ 1029]
	ResultSymbols.T. [▶ 2356]	Class representing the asynchronous result of reading a symbol enumeration of type IEnumerable.T. via ADS. Implements the ResultValue.TValue. [▶ 1029]
	RpcInvokeException [▶ 2359]	Class RpcInvokeException. Implements the SymbolException [▶ 2401]
	RpcMethodCollection [▶ 2364]	Collection of RpcMethods. [▶ 2123]
	RpcMethodNotSupportedException [▶ 2379]	Symbol Exception
	RpcMethodParameterCollection [▶ 2384]	Collection of RPC method parameters
	SymbolCollection [▶ 2396]	Interface represents a collection of ISymbol [▶ 2176] objects.
	SymbolException [▶ 2401]	Symbol bound exceptions
	TypeAttribute [▶ 2413]	ADS Attribute
	TypeAttributeCollection [▶ 2418]	Collection of AdsAttributes [▶ 2209]
	ValueChangedBaseEventArgs [▶ 2435]	Event args for the RawValueChanged [▶ 2254] event.
	ValueChangedEventArgs [▶ 2439]	Event args for the ValueChanged [▶ 2269] event.

Interfaces

	Interface	Description
	IAliasInstance [▶ 1952]	Interface representing an instance of an IAliasType [▶ 1954].
	IAliasType [▶ 1954]	Interface representing an Alias Type
	IAnyTypeMarshaler [▶ 1957]	Interface IAnyTypeMarshaler Implements the IGenericTypeMarshaler [▶ 2046]
	IArrayInstance [▶ 1964]	Interface representing an array instance
	IArrayType [▶ 1971]	Interface representing an array DataType [▶ 1986].
	IArrayValue [▶ 1976]	Interface IArrayValue
	IAttributedInstance [▶ 1980]	Interface IAttributedInstance
	IBitSize [▶ 1982]	Interface IBitSize
	IDataType [▶ 1986]	Base interface for objects representing data types
	IDataTypeCollection [▶ 1993]	Interface IDataTypeCollection Implements the ICollection.T.
	IDataTypeCollection.T. [▶ 1995]	Data Type container interface
	IDimension [▶ 1998]	Interface representing a single Dimension [▶ 1998] of an ArrayType [▶ 1971].
	IDimensionCollection [▶ 2000]	Interface IDimensionCollection
	IDynamicSymbol [▶ 2004]	Interface IDynamicSymbol
	IDynamicSymbolLoader [▶ 2007]	Dynamic symbol loader interface
	IDynamicSymbolsCollection [▶ 2010]	Interface IDynamicSymbolsContainer Implements the IDynamicMetaObjectProvider
	IDynamicValue [▶ 2011]	Interface IDynamicValue Implements the IDynamicMetaObjectProvider Implements the IValue [▶ 2226] Implements the IStructValue [▶ 2167] Implements the IArrayValue [▶ 1976]
	IEnumType [▶ 2014]	Common Enum type interface
	IEnumType.T. [▶ 2021]	Interface representing an enum type
	IEnumValue [▶ 2028]	Generic interface for EnumValues
	IEnumValueCollection [▶ 2031]	Interface IEnumValueCollection

	Interface	Description
	IEnumValueCollection.TEnumValue, TValue. [▶ 2033]	Interface for collections of IEnumValues [▶ 2028]. Implements the ICollection.T.
	IField [▶ 2040]	Specifies a single field/member of a Struct DataType [▶ 2162].
	IFieldCollection [▶ 2042]	Interface IFieldCollection Implements the IInstanceCollection.T. [▶ 2057]
	IGenericTypeMarshaller [▶ 2046]	Interface IGenericTypeMarshaller Implements the ITypeMarshaler [▶ 2216]
	IHierarchicalSymbol [▶ 2048]	Bindable Symbol interface (for internal use only)
	IInstance [▶ 2052]	Interface specifying instance objects.
	IInstanceCollection.T. [▶ 2057]	Generic InstanceCollection interface.
	IMember [▶ 2065]	Specifies a single field/member of a Struct DataType [▶ 2162].
	IMemberCollection [▶ 2068]	Interface IMemberCollection Implements the IInstanceCollection.T. [▶ 2057]
	INamespaceCollection [▶ 2072]	Interface INamespaceCollection
	INamespaceCollection.T. [▶ 2073]	Interface INamespaceCollection
	IOversamplingArrayInstance [▶ 2079]	Interface IOversamplingArrayInstance
	IPointerInstance [▶ 2083]	Interface representing an instance of an IPointerType [▶ 2086]
	IPointerType [▶ 2086]	Interface representing a pointer type
	IPrimitiveType [▶ 2089]	Interface IPrimitiveType
	IProcessImageAddress [▶ 2092]	Interface describing a Process Image Address
	IReferenceInstance [▶ 2094]	Interface representing an instance of an IReferenceType [▶ 2099]
	IReferenceType [▶ 2099]	Interface representing a reference/pointer type
	IRpcCallableInstance [▶ 2104]	Interface for an RPC callable PLC Method (Remote procedure call)
	IRpcCallableType [▶ 2121]	Interface representing an RPC callable IStructType [▶ 2162]
	IRpcMethod [▶ 2123]	Interface describes an RPC Method

	Interface	Description
	IRpcMethodCollection [▶ 2127]	Interface for RPC Method collections.
	IRpcMethodParameter [▶ 2133]	Interface IRpcMethodParameter
	IRpcMethodParameterCollection [▶ 2137]	Interface IRpcMethodParameterCollection
	IRpcStructInstance [▶ 2140]	Interface IRpcStructInstance
	IStringInstance [▶ 2145]	Interface IStringInstance
	IStringMarshaler [▶ 2148]	Common interface for marshalling ADS string values.
	IStringType [▶ 2155]	Interface representing a string IDataType [▶ 1986]
	IStructInstance [▶ 2158]	Interface representing an instance of a IStructType [▶ 2162]
	IStructType [▶ 2162]	Interface representing Struct data types
	IStructValue [▶ 2167]	Interface IStructValue
	ISubRangeType [▶ 2170]	Interface representing a SubRange type
	ISubRangeType.T. [▶ 2173]	Interface representing a SubRange type
	ISymbol [▶ 2176]	Interface specifying Symbols (
	ISymbolCollection [▶ 2182]	Interface ISymbolCollection Implements the IInstanceCollection.T. [▶ 2057]
	ISymbolCollection.T. [▶ 2185]	Interface ISymbolCollection
	ISymbolFactory [▶ 2188]	Symbol Factory Interface
	ISymbolFactoryServicesProvider [▶ 2198]	Symbol Value Access interface
	ISymbolInfo [▶ 2199]	Interface ISymbolInfo
	ISymbolLoader [▶ 2200]	Symbol Loader interface
	ISymbolProvider [▶ 2203]	Symbol Provider interface.
	ISymbolServer [▶ 2205]	Symbol Server Interface
	ITypeAttribute [▶ 2209]	Interface for ADS attributes

	Interface	Description
	ITypeAttributeCollection [▶ 2211]	Interface ITypeAttributeCollection
	ITypeMarshaler [▶ 2216]	Interface ITypeMarshaler
	IUnionInstance [▶ 2220]	Interface for an Instance of the IUnionType [▶ 2224].
	IUnionType [▶ 2224]	Interface for an union data type.
	IValue [▶ 2226]	Symbol Value Interface
	IValueAccessorProvider [▶ 2234]	Interface IValueAccessorProvider
	IValueAnySymbol [▶ 2235]	Interface IValueAnySymbol
	IValueRawSymbol [▶ 2244]	Interface IValueRawSymbol
	IValueSymbol [▶ 2254]	Interface for a ISymbol [▶ 2176] that supports values.
	IVirtualStructInstance [▶ 2269]	Virtual Struct instance interface.

Enumerations

	Enumeration	Description
	DataTypeCategory [▶ 1649]	Category of a DataType / Instance
	InstanceCollectionMode [▶ 2075]	Enum InstanceCollectionMode
	MethodParamFlags [▶ 2288]	Flag set specifying the MethodParameter context
	PrimitiveTypeFlags [▶ 2289]	Enum PrimitiveTypeFlags
	StringConvertMode [▶ 2395]	Enum StringConvertMode
	SymbolAccessRights [▶ 2396]	Enum specifying Access Rights to symbols

6.11.1 AnySymbolSpecifier Class

Class AnySymbolSpecifier.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.AnySymbolSpecifier
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AnySymbolSpecifier
```

The AnySymbolSpecifier type exposes the following members.

Constructors

	Name	Description
	AnySymbolSpecifier [▶ 1631]	Initializes a new instance of the AnySymbolSpecifier class.

Properties

	Name	Description
	InstancePath [▶ 1631]	Gets the instance path.
	TypeSpecifier [▶ 1632]	Gets the type specifier.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

Specifies an Symbol path together with an AnyTypeSpecifier [▶ 1633] to address a symbol for an Read/Write operation

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.1.1 AnySymbolSpecifier Constructor

Initializes a new instance of the AnySymbolSpecifier [▶ 1629] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnySymbolSpecifier(
    string instancePath,
    AnyTypeSpecifier spec
)
```

Parameters

instancePath	Type: System.String The instance path.
spec	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The spec.

Reference

AnySymbolSpecifier Class [▶ 1629]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.1.2 AnySymbolSpecifier Properties

The AnySymbolSpecifier [▶ 1629] type exposes the following members.

Properties

	Name	Description
	InstancePath [▶ 1631]	Gets the instance path.
	TypeSpecifier [▶ 1632]	Gets the type specifier.

Reference

AnySymbolSpecifier Class [▶ 1629]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.1.2.1 AnySymbolSpecifier.InstancePath Property

Gets the instance path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Reference

AnySymbolSpecifier Class [▶ 1629]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.1.2.2 AnySymbolSpecifier.TypeSpecifier Property

Gets the type specifier.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier TypeSpecifier { get; }
```

Property Value

Type: AnyTypeSpecifier [▶ 1633]
The type specifier.

Reference

AnySymbolSpecifier Class [▶ 1629]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.1.3 AnySymbolSpecifier Methods

The AnySymbolSpecifier [▶ 1629] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AnySymbolSpecifier Class [▶ 1629]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2 AnyTypeSpecifier Class

Class AnyTypeSpecifier.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.AnyTypeSpecifier
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class AnyTypeSpecifier
```

The AnyTypeSpecifier type exposes the following members.

Constructors

	Name	Description
	AnyTypeSpecifier(Object) [▶ 1635]	Initializes a new instance of the AnyTypeSpecifier class from the specified prototype
	AnyTypeSpecifier(Type) [▶ 1635]	Initializes a new instance of the AnyTypeSpecifier class.
	AnyTypeSpecifier(Type, IList.IDimensionCollection.) [▶ 1636]	Initializes a new instance of the AnyTypeSpecifier class.
	AnyTypeSpecifier(Type, Int32) [▶ 1636]	Initializes a new instance of the AnyTypeSpecifier class.
	AnyTypeSpecifier(Type, .Int32.) [▶ 1637]	Initializes a new instance of the AnyTypeSpecifier class.

Properties

	Name	Description
	Category [▶ 1638]	Category of the AnyTypeSpecifier
	DimLengths [▶ 1639]	List of jagged Dimensions (Arrays and jagged arrays)
	ElementType [▶ 1639]	The element type (Arrays)
	StrLen [▶ 1639]	The String length (only for String [▶ 1649]
	Type [▶ 1640]	Managed type

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

The AnyTypeSpecifier is used to specify out/return parameters for ReadAny, InvokeRpc-Methods or typed AdsNotifications.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1 AnyTypeSpecifier Constructor

Overload List

	Name	Description
	AnyTypeSpecifier(Object) [▶ 1635]	Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class from the specified prototype
	AnyTypeSpecifier(Type) [▶ 1635]	Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.
	AnyTypeSpecifier(Type, IList.IDimensionCollection.) [▶ 1636]	Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

	Name	Description
	AnyTypeSpecifier(Type, Int32) [▶ 1636]	Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.
	AnyTypeSpecifier(Type, .Int32.) [▶ 1637]	Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1.1 AnyTypeSpecifier Constructor (Object)

Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class from the specified prototype

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier(
    Object prototype
)
```

Parameters

prototype	Type: System.Object The prototype.

Reference

AnyTypeSpecifier Class [▶ 1633]

AnyTypeSpecifier Overload [▶ 1634]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1.2 AnyTypeSpecifier Constructor (Type)

Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier(
    Type type
)
```

Parameters

type	Type: System.Type The type.

Reference

AnyTypeSpecifier Class [▶ 1633]

AnyTypeSpecifier Overload [▶ 1634]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1.3 AnyTypeSpecifier Constructor (Type, IList.IDimensionCollection.)

Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier(
    Type type,
    IList<IDimensionCollection> dimLengths
)
```

Parameters

type	Type: System.Type The type.
dimLengths	Type: System.Collections.Generic.IList.IDimensionCollection [▶ 2000]. The dim lengths.

Reference

AnyTypeSpecifier Class [▶ 1633]

AnyTypeSpecifier Overload [▶ 1634]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1.4 AnyTypeSpecifier Constructor (Type, Int32)

Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier(
    Type type,
    int strLen
)
```

Parameters

type	Type: System.Type The type.
strLen	Type: System.Int32 Length of the string.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	type

Reference

AnyTypeSpecifier Class [▶ 1633]

AnyTypeSpecifier Overload [▶ 1634]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.1.5 AnyTypeSpecifier Constructor (Type, .Int32.)

Initializes a new instance of the AnyTypeSpecifier [▶ 1633] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier(
    Type type,
    int[] dimLengths
)
```

Parameters

type	Type: System.Type The type.
dimLengths	Type: .System.Int32. The dim lengths.

Exceptions

Exception	Condition
ArgumentOutOfRangeException	type

Reference

AnyTypeSpecifier Class [▶ 1633]

AnyTypeSpecifier Overload [▶ 1634]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2 AnyTypeSpecifier Properties

The AnyTypeSpecifier [▶ 1633] type exposes the following members.

Properties

	Name	Description
	Category [▶ 1638]	Category of the AnyTypeSpecifier [▶ 1633]
	DimLengths [▶ 1639]	List of jagged Dimensions (Arrays and jagged arrays)
	ElementType [▶ 1639]	The element type (Arrays)
	StrLen [▶ 1639]	The String length (only for String [▶ 1649]
	Type [▶ 1640]	Managed type

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2.1 AnyTypeSpecifier.Category Property

Category of the AnyTypeSpecifier [▶ 1633]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory Category { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2.2 AnyTypeSpecifier.DimLengths Property

List of jagged Dimensions (Arrays and jagged arrays)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IList<IDimensionCollection> DimLengths { get; }
```

Property Value

Type: IList.IDimensionCollection [▶ 2000].

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2.3 AnyTypeSpecifier.ElementType Property

The element type (Arrays)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public AnyTypeSpecifier ElementType { get; }
```

Property Value

Type: AnyTypeSpecifier [▶ 1633]

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2.4 AnyTypeSpecifier.StrLen Property

The String length (only for String [▶ 1649]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int StrLen { get; }
```

Property Value

Type: Int32

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.2.5 AnyTypeSpecifier.Type Property

Managed type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Type Type { get; }
```

Property Value

Type: Type

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.2.3 AnyTypeSpecifier Methods

The AnyTypeSpecifier [▶ 1633] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

AnyTypeSpecifier Class [▶ 1633]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.3 CannotAccessVirtualSymbolException Class

Cannot access virtual Symbol

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]
        TwinCAT.TypeSystem.CannotAccessVirtualSymbolException

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class CannotAccessVirtualSymbolException : SymbolException
```

The CannotAccessVirtualSymbolException type exposes the following members.

Constructors

	Name	Description
	CannotAccessVirtualSymbolException [▶ 1642]	Initializes a new instance of the CannotAccessVirtualSymbolException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.3.1 CannotAccessVirtualSymbolException Constructor

Initializes a new instance of the CannotAccessVirtualSymbolException [▶ 1641] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public CannotAccessVirtualSymbolException(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

CannotAccessVirtualSymbolException Class [▶ 1641]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.3.2 CannotAccessVirtualSymbolException Properties

The CannotAccessVirtualSymbolException [▶ 1641] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

CannotAccessVirtualSymbolException Class [▶ 1641]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.3.3 CannotAccessVirtualSymbolException Methods

The CannotAccessVirtualSymbolException [▶ 1641] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

CannotAccessVirtualSymbolException Class [▶ 1641]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4 CannotResolveDataTypeException Class

Class CannotResolveDataTypeException. Implements the DataTypeException [▶ 1658]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.DataTypeException [▶ 1658]
        TwinCAT.TypeSystem.CannotResolveDataTypeException

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class CannotResolveDataTypeException : DataTypeException
```

The CannotResolveDataTypeException type exposes the following members.

Constructors

	Name	Description
	CannotResolveDataTypeException(String) [▶ 1646]	Initializes a new instance of the CannotResolveDataTypeException class.
	CannotResolveDataTypeException(IInstance) [▶ 1647]	Initializes a new instance of the CannotResolveDataTypeException class.
	CannotResolveDataTypeException(SerializationInfo, StreamingContext) [▶ 1647]	Initializes a new instance of the CannotResolveDataTypeException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	DataType [▶ 1663]	Gets the type of the data. (Inherited from DataTypeException [▶ 1658].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)
	TypeName [▶ 1663]	Gets the name of the type. (Inherited from DataTypeException [▶ 1658].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1664]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from DataTypeException [▶ 1658].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DataTypeException [▶ 1658]

6.11.4.1 CannotResolveDataTypeException Constructor

Overload List

	Name	Description
	CannotResolveDataTypeException(String) [▶ 1646]	Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.
	CannotResolveDataTypeException(IInstance) [▶ 1647]	Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.
	CannotResolveDataTypeException(SerializationInfo, StreamingContext) [▶ 1647]	Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.

Reference

CannotResolveDataTypeException Class [▶ 1644]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.1.1 CannotResolveDataTypeException Constructor (String)

Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public CannotResolveDataTypeException(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the type.

Reference

CannotResolveDataTypeException Class [▶ 1644]

CannotResolveDataTypeException Overload [▶ 1646]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.1.2 CannotResolveDataTypeException Constructor (IInstance)

Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public CannotResolveDataTypeException(
    IInstance symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IInstance [▶ 2052] The symbol.

Reference

CannotResolveDataTypeException Class [▶ 1644]

CannotResolveDataTypeException Overload [▶ 1646]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.1.3 CannotResolveDataTypeException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the CannotResolveDataTypeException [▶ 1644] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected CannotResolveDataTypeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

CannotResolveDataTypeException Class [▶ 1644]

CannotResolveDataTypeException Overload [▶ 1646]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.2 CannotResolveDataTypeException Properties

The CannotResolveDataTypeException [▶ 1644] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	DataType [▶ 1663]	Gets the type of the data. (Inherited from DataTypeException [▶ 1658].)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)
	TypeName [▶ 1663]	Gets the name of the type. (Inherited from DataTypeException [▶ 1658].)

Reference

CannotResolveDataTypeException Class [▶ 1644]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.3 CannotResolveDataTypeException Methods

The CannotResolveDataTypeException [▶ 1644] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1664]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from DataTypeException [▶ 1658].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

CannotResolveDataTypeException Class [▶ 1644]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.4.4 CannotResolveDataTypeException Events

The CannotResolveDataTypeException [▶ 1644] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

CannotResolveDataTypeException Class [▶ 1644]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.5 DataTypeCategory Enumeration

Category of a DataType / Instance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum DataTypeCategory
```

Members

	Member name	Value	Description
	None	0	Uninitialized / NotProcessed (0)

	Member name	Value	Description
	Primitive	1	Simple / Base Data Type (1)
	Alias	2	Alias data type (2)
	Enum	3	Enumeration data type (3)
	Array	4	Array data type (4)
	Struct	5	Structure data type (5)
	FunctionBlock	6	Function block (POU) (6)
	Program	7	Program (POU) (7)
	Function	8	Function (POU) (8)
	SubRange	9	SubRange (9)
	String	10	Fixed length string (10)
	Bitset	12	Bitset (12)
	Pointer	13	Pointer type (13)
	Union	14	Union type (14)
	Reference	15	Reference type (15)
	Interface	16	The interface

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6 DataTypeCollection Class

Collection of DataTypes. [▶ 1986]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.DataTypeCollection [▶ 2442].IDataType [▶ 1986].
    TwinCAT.TypeSystem.DataTypeCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DataTypeCollection : DataTypeCollection<IDataType>,
    IDataTypeCollection, IDataTypeCollection<IDataType>, ICollection<IDataType>,
    IEnumerable<IDataType>, IEnumerable
```

The DataTypeCollection type exposes the following members.

Constructors

	Name	Description
	DataTypeCollection. [▶ 1652]	Initializes a new instance of the DataTypeCollection class.
	DataTypeCollection(IEnumerable.IDataType.) [▶ 1653]	Initializes a new instance of the DataTypeCollection class (Copy constructor).

Properties

	Name	Description
	Count [▶ 2445]	Gets the count of contained IDataType [▶ 1986]s. (Inherited from DataTypeCollection.T. [▶ 2442].)
	IsReadOnly [▶ 2446]	Gets a value indicating whether this instance is read only. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Item.Int32. [▶ 2447]	Gets or sets the IDataType [▶ 1986] at the specified index. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Item.String. [▶ 2448]	Gets the IDataType [▶ 1986] with the specified name. (Inherited from DataTypeCollection.T. [▶ 2442].)

Methods

	Name	Description
	Add [▶ 2449]	Adds the specified item to the collection. (Inherited from DataTypeCollection.T. [▶ 2442].)
	AddRange [▶ 2450]	Adds a range of types (Inherited from DataTypeCollection.T. [▶ 2442].)
	AsReadOnly [▶ 1654]	Returns A ReadOnly-Version of the DataTypeCollection.
	Clear [▶ 2451]	Clears the collection. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Clone [▶ 1655]	Clones this DataTypeCollection (Shallow Copy)
	Contains [▶ 2451]	Determines whether this DataTypeCollection contains the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	ContainsType [▶ 2452]	Determines whether the container contains the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	CopyTo [▶ 2453]	Copies the data types to the specified array, starting at the array index. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2453]	Gets the enumerator. (Inherited from DataTypeCollection.T. [▶ 2442].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2454]	Determines the Index of the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Insert [▶ 2454]	Inserts an IDataType [▶ 1986] into the DataTypeCollection. (Inherited from DataTypeCollection.T. [▶ 2442].)
	LookupType [▶ 2455]	Determines the specified IDataType [▶ 1986] (Inherited from DataTypeCollection.T. [▶ 2442].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2456]	Removes the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	RemoveAt [▶ 2456]	Removes the IDataType [▶ 1986] object at the specified index. (Inherited from DataTypeCollection.T. [▶ 2442].)

	Name	Description
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2457]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995]. (Inherited from DataTypeCollection.T. [▶ 2442].)

Fields

	Name	Description
	list [▶ 2458]	Internal list of data types (Inherited from DataTypeCollection.T. [▶ 2442].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.1 DataTypeCollection Constructor

Overload List

	Name	Description
	DataTypeCollection. [▶ 1652]	Initializes a new instance of the DataTypeCollection [▶ 1650] class.
	DataTypeCollection(IEnumerable.IDataType.) [▶ 1653]	Initializes a new instance of the DataTypeCollection [▶ 1650] class (Copy constructor).

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.1.1 DataTypeCollection Constructor

Initializes a new instance of the DataTypeCollection [▶ 1650] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection()
```

Reference

DataTypeCollection Class [▶ 1650]

DataTypeCollection Overload [▶ 1652]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.1.2 DataTypeCollection Constructor (IEnumerable.IDataType.)

Initializes a new instance of the DataTypeCollection [▶ 1650] class (Copy constructor).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection(
    IEnumerable<IDataType> coll
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.IDataType [▶ 1986]. The coll.

Reference

DataTypeCollection Class [▶ 1650]

DataTypeCollection Overload [▶ 1652]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.2 DataTypeCollection Properties

The DataTypeCollection [▶ 1650] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2445]	Gets the count of contained IDataType [▶ 1986]s. (Inherited from DataTypeCollection.T. [▶ 2442].)
	IsReadOnly [▶ 2446]	Gets a value indicating whether this instance is read only. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Item.Int32. [▶ 2447]	Gets or sets the IDataType [▶ 1986] at the specified index. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Item.String. [▶ 2448]	Gets the IDataType [▶ 1986] with the specified name. (Inherited from DataTypeCollection.T. [▶ 2442].)

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.3 DataTypeCollection Methods

The DataTypeCollection [▶ 1650] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2449]	Adds the specified item to the collection. (Inherited from DataTypeCollection.T. [▶ 2442].)
	AddRange [▶ 2450]	Adds a range of types (Inherited from DataTypeCollection.T. [▶ 2442].)
	AsReadOnly [▶ 1654]	Returns A ReadOnly-Version of the DataTypeCollection [▶ 1650].
	Clear [▶ 2451]	Clears the collection. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Clone [▶ 1655]	Clones this DataTypeCollection [▶ 1650] (Shallow Copy)
	Contains [▶ 2451]	Determines whether this DataTypeCollection [▶ 1650] contains the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	ContainsType [▶ 2452]	Determines whether the container contains the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	CopyTo [▶ 2453]	Copies the data types to the specified array, starting at the array index. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2453]	Gets the enumerator. (Inherited from DataTypeCollection.T. [▶ 2442].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2454]	Determines the Index of the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	Insert [▶ 2454]	Inserts an IDataType [▶ 1986] into the DataTypeCollection [▶ 1650]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	LookupType [▶ 2455]	Determines the specified IDataType [▶ 1986] (Inherited from DataTypeCollection.T. [▶ 2442].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2456]	Removes the specified IDataType [▶ 1986]. (Inherited from DataTypeCollection.T. [▶ 2442].)
	RemoveAt [▶ 2456]	Removes the IDataType [▶ 1986] object at the specified index. (Inherited from DataTypeCollection.T. [▶ 2442].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2457]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995]. (Inherited from DataTypeCollection.T. [▶ 2442].)

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.3.1 DataTypeCollection.AsReadOnly Method

Returns A ReadOnly-Version of the DataTypeCollection [▶ 1650].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDataTypeCollection AsReadOnly()
```

Return Value

Type: ReadOnlyDataTypeCollection [▶ 2291]
A read only version of this DataTypeCollection [▶ 1650].

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.3.2 DataTypeCollection.Clone Method

Clones this DataTypeCollection [▶ 1650] (Shallow Copy)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection Clone()
```

Return Value

Type: DataTypeCollection [▶ 1650]
A clone of this DataTypeCollection [▶ 1650].

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.6.4 DataTypeCollection Fields

The DataTypeCollection [▶ 1650] type exposes the following members.

Fields

	Name	Description
	list [▶ 2458]	Internal list of data types (Inherited from DataTypeCollection.T. [▶ 2442].)

Reference

DataTypeCollection Class [▶ 1650]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.7 DataTypeEventArgs Class

Class DataTypeEventArgs.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.TypeSystem.DataTypeEventArgs
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DataTypeEventArgs : EventArgs
```

The DataTypeEventArgs type exposes the following members.

Constructors

	Name	Description
	DataTypeEventArgs [▶ 1657]	Initializes a new instance of the DataTypeEventArgs class.

Properties

	Name	Description
	DataTypes [▶ 1657]	The data types

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.EventArgs

6.11.7.1 DataTypeEventArgs Constructor

Initializes a new instance of the DataTypeEventArgs [▶ 1656] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeEventArgs(
    IEnumerable<IDataType> types
)
```

Parameters

types	Type: System.Collections.Generic.IEnumerable.IDataType [▶ 1986]. The types.

Reference

DataTypeEventArgs Class [▶ 1656]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.7.2 DataTypeEventArgs Properties

The DataTypeEventArgs [▶ 1656] type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 1657]	The data types

Reference

DataTypeEventArgs Class [▶ 1656]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.7.2.1 DataTypeEventArgs.DataTypes Property

The data types

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerable<IDataType> DataTypes { get; }
```

Property Value

Type: IEnumerable.IDataType [▶ 1986].

Reference

DataTypeEventArgs Class [▶ 1656]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.7.3 DataTypeEventArgs Methods

The DataTypeEventArgs [▶ 1656] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

DataTypeEventArgs Class [▶ 1656]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8 DataTypeException Class

Data Type Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.DataTypeException
        TwinCAT.TypeSystem.CannotResolveDataTypeException [▶ 1644]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class DataTypeException : AdsException
```

The DataTypeException type exposes the following members.

Constructors

	Name	Description
	DataTypeException(SerializationInfo, StreamingContext) [▶ 1660]	Initializes a new instance of the DataTypeException class.
	DataTypeException(String, String) [▶ 1661]	Initializes a new instance of the DataTypeException class.
	DataTypeException(String, IDataType) [▶ 1662]	Initializes a new instance of the DataTypeException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	DataType [▶ 1663]	Gets the type of the data.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)
	TypeName [▶ 1663]	Gets the name of the type.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1664]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.1 DataTypeException Constructor

Overload List

	Name	Description
	DataTypeException(SerializationInfo, StreamingContext) [▶ 1660]	Initializes a new instance of the DataTypeException [▶ 1658] class.
	DataTypeException(String, String) [▶ 1661]	Initializes a new instance of the DataTypeException [▶ 1658] class.
	DataTypeException(String, IDataType) [▶ 1662]	Initializes a new instance of the DataTypeException [▶ 1658] class.

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.1.1 DataTypeException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the DataTypeException [▶ 1658] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected DataTypeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

DataTypeException Class [▶ 1658]

DataTypeException Overload [▶ 1660]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.1.2 DataTypeException Constructor (String, String)

Initializes a new instance of the DataTypeException [▶ 1658] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeException(
    string message,
    string type
)
```

Parameters

message	Type: System.String The message.
type	Type: System.String The type.

Reference

DataTypeException Class [▶ 1658]

DataTypeException Overload [▶ 1660]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.1.3 DataTypeException Constructor (String, IDataType)

Initializes a new instance of the DataTypeException [▶ 1658] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeException(
    string message,
    IDataType type
)
```

Parameters

message	Type: System.String The message.
type	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] The type.

Reference

DataTypeException Class [▶ 1658]

DataTypeException Overload [▶ 1660]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.2 DataTypeException Properties

The DataTypeException [▶ 1658] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	DataType [▶ 1663]	Gets the type of the data.
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

	Name	Description
	TypeName [▶ 1663]	Gets the name of the type.

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.2.1 DataTypeException.DataType Property

Gets the type of the data.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType DataType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.2.2 DataTypeException.TypeName Property

Gets the name of the type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string TypeName { get; }
```

Property Value

Type: String
The name of the type.

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.3 DataTypeException Methods

The DataTypeException [▶ 1658] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 1664]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.3.1 DataTypeException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
Exception.GetObjectData(SerializationInfo, StreamingContext)

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.8.4 DataTypeException Events

The DataTypeException [▶ 1658] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

DataTypeException Class [▶ 1658]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.9 DataTypeNameEventArgs Class

Class DataTypeNameEventArgs.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.TypeSystem.DataTypeNameEventArgs

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DataTypeNameEventArgs : EventArgs
```

The DataTypeNameEventArgs type exposes the following members.

Constructors

	Name	Description
	DataTypeNameEventArgs [▶ 1666]	Initializes a new instance of the DataTypeNameEventArgs class.

Properties

	Name	Description
	TypeName [▶ 1667]	The type name

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.EventArgs

6.11.9.1 DataTypeNameEventArgs Constructor

Initializes a new instance of the DataTypeNameEventArgs [▶ 1665] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeNameEventArgs(
    string typeName
)
```

Parameters

typeName	Type: System.String Name of the type.

Reference

DataTypeNameEventArgs Class [▶ 1665]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.9.2 DataTypeNameEventArgs Properties

The DataTypeNameEventArgs [▶ 1665] type exposes the following members.

Properties

	Name	Description
	TypeName [▶ 1667]	The type name

Reference

DataTypeNameEventArgs Class [▶ 1665]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.9.2.1 DataTypeNameEventArgs.TypeName Property

The type name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string TypeName { get; }
```

Property Value

Type: String

Reference

DataTypeNameEventArgs Class [▶ 1665]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.9.3 DataTypeNameEventArgs Methods

The DataTypeNameEventArgs [▶ 1665] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

DataTypeNameEventArgs Class [▶ 1665]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10 Dimension Class

Represents a single dimension of an IArrayType [▶ 1971]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Dimension
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class Dimension : IDimension
```

The Dimension type exposes the following members.

Properties

	Name	Description
	ElementCount [▶ 1669]	Gets the number of elements within that IDimension [▶ 1998].
	LowerBound [▶ 1669]	Gets the lower bound of elements within that IDimension [▶ 1998].
	UpperBound [▶ 1670]	Gets the upper bound of elements within this Dimension

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10.1 Dimension Properties

The Dimension [▶ 1668] type exposes the following members.

Properties

	Name	Description
	ElementCount [▶ 1669]	Gets the number of elements within that IDimension [▶ 1998].
	LowerBound [▶ 1669]	Gets the lower bound of elements within that IDimension [▶ 1998].
	UpperBound [▶ 1670]	Gets the upper bound of elements within this Dimension [▶ 1668]

Reference

Dimension Class [▶ 1668]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10.1.1 Dimension.ElementCount Property

Gets the number of elements within that IDimension [▶ 1998].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ElementCount { get; }
```

Property Value

Type: Int32
The element count.

Implements

IDimension.ElementCount [▶ 1999]

Reference

Dimension Class [▶ 1668]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10.1.2 Dimension.LowerBound Property

Gets the lower bound of elements within that IDimension [▶ 1998].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int LowerBound { get; }
```

Property Value

Type: Int32
The lower bound.

Implements

IDimension.LowerBound [▶ 1999]

Reference

Dimension Class [▶ 1668]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10.1.3 Dimension.UpperBound Property

Gets the upper bound of elements within this Dimension [▶ 1668]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int UpperBound { get; }
```

Property Value

Type: Int32
The upper bound.

Reference

Dimension Class [▶ 1668]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.10.2 Dimension Methods

The Dimension [▶ 1668] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

Dimension Class [▶ 1668]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11 DimensionCollection Class

Collection class for Array Dimensions

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.DimensionCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DimensionCollection : IDimensionCollection,
    IList<IDimension>, ICollection<IDimension>, IEnumerable<IDimension>,
    IEnumerable
```

The DimensionCollection type exposes the following members.

Constructors

	Name	Description
	DimensionCollection. [▶ 1673]	Initializes a new instance of the DimensionCollection class.
	DimensionCollection(Int32) [▶ 1673]	Initializes a new instance of an 1-Dimensional representing DimensionCollection class.
	DimensionCollection(.Int32.) [▶ 1674]	Initializes a new instance of the DimensionCollection class.
	DimensionCollection(IEnumerable.IDimension.) [▶ 1674]	Initializes a new instance of the DimensionCollection class.

Properties

	Name	Description
	Count [▶ 1675]	Gets the number of elements contained in the ICollection.T..
	ElementCount [▶ 1676]	Gets the Number of elements in all Dimensions
	IsReadOnly [▶ 1677]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 1677]	Gets or sets the element at the specified index.
	LowerBounds [▶ 1678]	Gets the lower bounds.
	UpperBounds [▶ 1678]	Gets the upper bounds.

Methods

	Name	Description
	Add [▶ 1680]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 1680]	Returns a read only version of this DimensionCollection.
	Clear [▶ 1681]	Removes all items from the ICollection.T..
	Contains [▶ 1682]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1682]	Copies to.
S	Empty [▶ 1683]	Gets an empty DimensionCollection
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetDimensionLengths [▶ 1683]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator [▶ 1684]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 1685]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1685]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 1686]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1687]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.1 DimensionCollection Constructor

Overload List

	Name	Description
	DimensionCollection. [▶ 1673]	Initializes a new instance of the DimensionCollection [▶ 1671] class.
	DimensionCollection(Int32) [▶ 1673]	Initializes a new instance of an 1-Dimensional representing DimensionCollection [▶ 1671] class.
	DimensionCollection(.Int32.) [▶ 1674]	Initializes a new instance of the DimensionCollection [▶ 1671] class.
	DimensionCollection(IEnumerable.IDimension.) [▶ 1674]	Initializes a new instance of the DimensionCollection [▶ 1671] class.

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.1.1 DimensionCollection Constructor

Initializes a new instance of the DimensionCollection [▶ 1671] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DimensionCollection()
```

Reference

DimensionCollection Class [▶ 1671]

DimensionCollection Overload [▶ 1673]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.1.2 DimensionCollection Constructor (Int32)

Initializes a new instance of an 1-Dimensional representing DimensionCollection [▶ 1671] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DimensionCollection(
    int length
)
```

Parameters

length	Type: System.Int32 The length.

Reference

DimensionCollection Class [▶ 1671]

DimensionCollection Overload [▶ 1673]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.1.3 DimensionCollection Constructor (.Int32.)

Initializes a new instance of the DimensionCollection [▶ 1671] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DimensionCollection(
    int[] dimLengths
)
```

Parameters

dimLengths	Type: .System.Int32. The dim lengths.

Reference

DimensionCollection Class [▶ 1671]

DimensionCollection Overload [▶ 1673]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.1.4 DimensionCollection Constructor (IEnumerable.IDimension.)

Initializes a new instance of the DimensionCollection [▶ 1671] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DimensionCollection(
    IEnumerable<IDimension> coll
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.IDimension [▶ 1998]. The coll.

Reference

DimensionCollection Class [▶ 1671]

DimensionCollection Overload [▶ 1673]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2 DimensionCollection Properties

The DimensionCollection [▶ 1671] type exposes the following members.

Properties

	Name	Description
	Count [▶ 1675]	Gets the number of elements contained in the ICollection.T..
	ElementCount [▶ 1676]	Gets the Number of elements in all Dimensions
	IsReadOnly [▶ 1677]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 1677]	Gets or sets the element at the specified index.
	LowerBounds [▶ 1678]	Gets the lower bounds.
	UpperBounds [▶ 1678]	Gets the upper bounds.

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.1 DimensionCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.2 DimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ElementCount { get; }
```

Property Value

Type: Int32

Implements

IDimensionCollection.ElementCount [▶ 2002]

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.3 DimensionCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.4 DimensionCollection.Item Property

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDimension this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: IDimension [▶ 1998]
IDimension.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Remarks

Index = 0 is the lowest dimesion.

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.5 DimensionCollection.LowerBounds Property

Gets the lower bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] LowerBounds { get; }
```

Property Value

Type: .Int32.
The lower bounds.

Implements

IDimensionCollection.LowerBounds [▶ 2002]

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.2.6 DimensionCollection.UpperBounds Property

Gets the upper bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] UpperBounds { get; }
```

Property Value

Type: .Int32.
The upper bounds.

Implements

IDimensionCollection.UpperBounds [▶ 2002]

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3 DimensionCollection Methods

The DimensionCollection [▶ 1671] type exposes the following members.

Methods

	Name	Description
	Add [▶ 1680]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 1680]	Returns a read only version of this DimensionCollection [▶ 1671].
	Clear [▶ 1681]	Removes all items from the ICollection.T..
	Contains [▶ 1682]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1682]	Copies to.
S	Empty [▶ 1683]	Gets an empty DimensionCollection [▶ 1671]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetDimensionLengths [▶ 1683]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator [▶ 1684]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 1685]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1685]	Inserts an item to the IList.T. at the specified index.

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 1686]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1687]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.1 DimensionCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    IDimension item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IDimension [▶ 1998] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.2 DimensionCollection.AsReadOnly Method

Returns a read only version of this DimensionCollection [▶ 1671].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDimensionCollection AsReadOnly()
```

Field Value

Type: ReadOnlyDimensionCollection [▶ 2295]
As read only.

Return Value

Type: ReadOnlyDimensionCollection [▶ 2295]
ReadOnlyDimensionCollection.

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.3 DimensionCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.4 DimensionCollection.Contains Method

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    IDimension item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IDimension [▶ 1998] The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.5 DimensionCollection.CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    IDimension[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.IDimension [▶ 1998]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.6 DimensionCollection.Empty Method

Gets an empty DimensionCollection [▶ 1671]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DimensionCollection Empty()
```

Field Value

Type: DimensionCollection [▶ 1671]
The Empty collectio.

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.7 DimensionCollection.GetDimensionLengths Method

Gets an array the specifies the Lengths of each Array Dimension

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] GetDimensionLengths()
```

Return Value

Type: .Int32.
System.Int32[].

Implements

IDimensionCollection.GetDimensionLengths. [▶ 2004]

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.8 DimensionCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<IDimension> GetEnumerator()
```

Return Value

Type: IEnumerator.IDimension [▶ 1998].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.9 DimensionCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    IDimension item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IDimension [▶ 1998] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.10 DimensionCollection.Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    IDimension item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.IDimension [▶ 1998] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.11 DimensionCollection.Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    IDimension item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IDimension [▶ 1998] The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.11.3.12 DimensionCollection.RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Exceptions

Exception	Condition
NotImplementedException	

Reference

DimensionCollection Class [▶ 1671]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12 DynamicAliasInstance Class

Class DynamicAliasInstance. This class cannot be inherited.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicAliasInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicAliasInstance : DynamicSymbol,
    IAliasInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicAliasInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1694]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetIndex [▶ 1694]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1695]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1696]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TrySetMember [▶ 1697]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]

TwinCAT.TypeSystem.IAliasInstance [▶ 1952]

6.11.12.1 DynamicAliasInstance Properties

The DynamicAliasInstance [▶ 1687] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2 DynamicAliasInstance Methods

The DynamicAliasInstance [▶ 1687] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1694]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)

	Name	Description
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetIndex [▶ 1694]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1695]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1696]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TrySetMember [▶ 1697]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2.1 DynamicAliasInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2.2 DynamicAliasInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.
result	Type: System.Object. The result of the index operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2.3 DynamicAliasInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2.4 DynamicAliasInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[][] is equal to 3.
value	Type: System.Object The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.2.5 DynamicAliasInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
value	Type: System.Object The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.12.3 DynamicAliasInstance Events

The DynamicAliasInstance [▶ 1687] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicAliasInstance Class [▶ 1687]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13 DynamicArrayInstance Class

Dynamic Array Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicArrayInstance
      TwinCAT.TypeSystem.DynamicOversamplingArrayInstance [▶ 1712]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicArrayInstance : DynamicSymbol,
    IArrayInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicArrayInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	Dimensions [▶ 1704]	Gets the dimensions as read only collection.
	Elements [▶ 1704]	Gets the contained Array Elements as read only collection.
	ElementType [▶ 1705]	Gets the type of the contained elements.
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	Item [▶ 1705]	Gets the ISymbol [▶ 2176] with the specified indices.
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetElement(IList.. Int32.., ISymbol.) [▶ 1709]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1709]	Tries to get the array element
	TryGetIndex [▶ 1710]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1711]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.1 DynamicArrayInstance Properties

The DynamicArrayInstance [▶ 1698] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	Dimensions [▶ 1704]	Gets the dimensions as read only collection.
	Elements [▶ 1704]	Gets the contained Array Elements as read only collection.
	ElementType [▶ 1705]	Gets the type of the contained elements.
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	Item [▶ 1705]	Gets the ISymbol [▶ 2176] with the specified indices.
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.1.1 DynamicArrayInstance.Dimensions Property

Gets the dimensions as read only collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDimensionCollection Dimensions { get; }
```

Property Value

Type: IDimensionCollection [▶ 2000]
The dimensions.

Implements

IArrayInstance.Dimensions [▶ 1967]

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.1.2 DynamicArrayInstance.Elements Property

Gets the contained Array Elements as read only collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolCollection<ISymbol> Elements { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The elements.

Implements

IArrayInstance.Elements [▶ 1968]

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.1.3 DynamicArrayInstance.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ElementType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the element.

Implements

IArrayInstance.ElementType [▶ 1968]

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.1.4 DynamicArrayInstance.Item Property

Gets the ISymbol [▶ 2176] with the specified indices.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol this[
    int[] indices
] { get; }
```

Parameters

indices	Type: .System.Int32. The indices.

Return Value

Type: ISymbol [▶ 2176]
ISymbol.

Implements

IArrayInstance.Item..Int32.. [▶ 1969]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	indices

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.2 DynamicArrayInstance Methods

The DynamicArrayInstance [▶ 1698] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetElement(IList.., Int32.., ISymbol.) [▶ 1709]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1709]	Tries to get the array element
	TryGetIndex [▶ 1710]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	TrySetIndex [▶ 1711]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.2.1 DynamicArrayInstance.TryGetElement Method

Overload List

	Name	Description
	TryGetElement(IList.., Int32.., ISymbol.) [▶ 1709]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1709]	Tries to get the array element

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicArrayInstance.TryGetElement Method (IList..Int32.., ISymbol.)

Tries to get the array element with the specified indices (jagged array support).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetElement(
    IList<int[]> jaggedIndices,
    out ISymbol symbol
)
```

Parameters

jaggedIndices	Type: System.Collections.Generic.IList..Int32.. The jagged indices list.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]. The symbol.

Return Value

Type: Boolean
true if found, false if the jagged indices specifiers is out-of-range.

Implements

IArrayInstance.TryGetElement(IList..Int32.., ISymbol.) [▶ 1970]

Exceptions

Exception	Condition
ArgumentNullException	jaggedIndices
ArgumentOutOfRangeException	jaggedIndices

Reference

DynamicArrayInstance Class [▶ 1698]

TryGetElement Overload [▶ 1708]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicArrayInstance.TryGetElement Method (.Int32., ISymbol.)

Tries to get the array element

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetElement(
    int[] indices,
    out ISymbol symbol
)
```

Parameters

indices	Type: .System.Int32. The indices.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]. The found Array element symbol (out-parameter).

Return Value

Type: Boolean
true if found, false if the indices specifiers is out-of-range.

Implements

IArrayInstance.TryGetElement(.Int32., ISymbol.) [▶ 1971]

Reference

DynamicArrayInstance Class [▶ 1698]

TryGetElement Overload [▶ 1708]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.2.2 DynamicArrayInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.

result	Type: System.Object. The result of the index operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.2.3 DynamicArrayInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[][] is equal to 3.
value	Type: System.Object The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.3 DynamicArrayInstance Events

The DynamicArrayInstance [▶ 1698] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.13.4 DynamicArrayInstance Fields

The DynamicArrayInstance [▶ 1698] type exposes the following members.

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicArrayInstance Class [▶ 1698]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.14 DynamicOversamplingArrayInstance Class

Dynamic Array Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicArrayInstance [▶ 1698]
      TwinCAT.TypeSystem.DynamicOversamplingArrayInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicOversamplingArrayInstance : DynamicArrayInstance,
    IOversamplingArrayInstance, IArrayInstance, ISymbol, IAttributedInstance, IInstance,
    IBitSize
```

The DynamicOversamplingArrayInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	Dimensions [▶ 1704]	Gets the dimensions as read only collection. (Inherited from DynamicArrayInstance [▶ 1698].)
	Elements [▶ 1704]	Gets the contained Array Elements as read only collection. (Inherited from DynamicArrayInstance [▶ 1698].)
	ElementType [▶ 1705]	Gets the type of the contained elements. (Inherited from DynamicArrayInstance [▶ 1698].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	Item [▶ 1705]	Gets the ISymbol [▶ 2176] with the specified indices. (Inherited from DynamicArrayInstance [▶ 1698].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	OversamplingElement [▶ 1717]	Gets the oversampling element.
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetElement(IList.., Int32.., ISymbol.) [▶ 1709]	Tries to get the array element with the specified indices (jagged array support). (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetElement(.Int32., ISymbol.) [▶ 1709]	Tries to get the array element (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetIndex [▶ 1710]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1711]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicArrayInstance [▶ 1698].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.14.1 DynamicOversamplingArrayInstance Properties

The DynamicOversamplingArrayInstance [▶ 1712] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	Dimensions [▶ 1704]	Gets the dimensions as read only collection. (Inherited from DynamicArrayInstance [▶ 1698].)
	Elements [▶ 1704]	Gets the contained Array Elements as read only collection. (Inherited from DynamicArrayInstance [▶ 1698].)
	ElementType [▶ 1705]	Gets the type of the contained elements. (Inherited from DynamicArrayInstance [▶ 1698].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	Item [▶ 1705]	Gets the ISymbol [▶ 2176] with the specified indices. (Inherited from DynamicArrayInstance [▶ 1698].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	OversamplingElement [▶ 1717]	Gets the oversampling element.
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicOversamplingArrayInstance Class [▶ 1712]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.14.1.1 DynamicOversamplingArrayInstance.OversamplingElement Property

Gets the oversampling element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol OversamplingElement { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The oversampling element.

Implements

IOversamplingArrayInstance.OversamplingElement [▶ 2082]

Exceptions

Exception	Condition
NotImplementedException	

Reference

DynamicOversamplingArrayInstance Class [▶ 1712]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.14.2 DynamicOversamplingArrayInstance Methods

The DynamicOversamplingArrayInstance [▶ 1712] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetElement(IList.. Int32.., ISymbol.) [▶ 1709]	Tries to get the array element with the specified indices (jagged array support). (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetElement(.Int32., ISymbol.) [▶ 1709]	Tries to get the array element (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetIndex [▶ 1710]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicArrayInstance [▶ 1698].)
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1711]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicArrayInstance [▶ 1698].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicOversamplingArrayInstance Class [▶ 1712]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.14.3 DynamicOversamplingArrayInstance Events

The DynamicOversamplingArrayInstance [▶ 1712] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicOversamplingArrayInstance Class [▶ 1712]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15 DynamicPointerInstance Class

Dynamic Pointer Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicPointerInstance
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicPointerInstance : DynamicSymbol,
    IPointerInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicPointerInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Reference [▶ 1725]	Gets the resolved reference of Pointer / Reference
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1727]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1727]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.1 DynamicPointerInstance Properties

The DynamicPointerInstance [▶ 1720] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Reference [▶ 1725]	Gets the resolved reference of Pointer / Reference

	Name	Description
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.1.1 DynamicPointerInstance.Reference Property

Gets the resolved reference of Pointer / Reference

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Reference { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The reference symbol or NULL if PVOID Pointer.

Implements

IPointerInstance.Reference [▶ 2086]

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.2 DynamicPointerInstance Methods

The DynamicPointerInstance [▶ 1720] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	GetDynamicMemberNames [▶ 1727]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1727]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.2.1 DynamicPointerInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.2.2 DynamicPointerInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.15.3 DynamicPointerInstance Events

The DynamicPointerInstance [▶ 1720] type exposes the following members.

Events

	Name	Description
⚡	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
⚡	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicPointerInstance Class [▶ 1720]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.16 DynamicPointerValue Class

Class DynamicPointerValue.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicValue [▶ 1869]
    TwinCAT.TypeSystem.DynamicPointerValue
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicPointerValue : DynamicValue
```

The DynamicPointerValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value) (Inherited from DynamicValue [▶ 1869].)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data. (Inherited from DynamicValue [▶ 1869].)
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226] (Inherited from DynamicValue [▶ 1869].)
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from DynamicValue [▶ 1869].)
	ResolvedType [▶ 1875]	Gets the resolved type. (Inherited from DynamicValue [▶ 1869].)
	Symbol [▶ 1875]	Gets the symbol that is bound to this value. (Inherited from DynamicValue [▶ 1869].)
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue [▶ 1869].)
	UpdateMode [▶ 1876]	Gets / Sets the update mode (Inherited from DynamicValue [▶ 1869].)
	ValueFactory [▶ 1877]	The value factory (Inherited from DynamicValue [▶ 1869].)

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1733]	Returns the enumeration of all dynamic member names. (Overrides DynamicValue.GetDynamicMemberNames. [▶ 1879].)
	Read [▶ 1879]	Reads the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	ReadAsync [▶ 1880]	read as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	ReadMember [▶ 1880]	Reads the specified member element. (Inherited from DynamicValue [▶ 1869].)
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	ToString [▶ 1882]	Returns a String that represents this instance. (Inherited from DynamicValue [▶ 1869].)
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue [▶ 1869].)
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value. (Inherited from DynamicValue [▶ 1869].)
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TryGetMemberValue [▶ 1733]	Tries the get member value. (Overrides DynamicValue.TryGetMemberValue(String, Object.) [▶ 1887].)
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue [▶ 1869].)
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue [▶ 1869].)
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays (Inherited from DynamicValue [▶ 1869].)
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value (Inherited from DynamicValue [▶ 1869].)
	Write [▶ 1893]	Writes the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	WriteAsync [▶ 1893]	write as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	WriteMember [▶ 1894]	Writes the specified member element. (Inherited from DynamicValue [▶ 1869].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DynamicValue [▶ 1869]

6.11.16.1 DynamicPointerValue Properties

The DynamicPointerValue [▶ 1728] type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value) (Inherited from DynamicValue [▶ 1869].)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data. (Inherited from DynamicValue [▶ 1869].)
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226] (Inherited from DynamicValue [▶ 1869].)
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from DynamicValue [▶ 1869].)
	ResolvedType [▶ 1875]	Gets the resolved type. (Inherited from DynamicValue [▶ 1869].)
	Symbol [▶ 1875]	Gets the symbol that is bound to this value. (Inherited from DynamicValue [▶ 1869].)
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue [▶ 1869].)
	UpdateMode [▶ 1876]	Gets / Sets the update mode (Inherited from DynamicValue [▶ 1869].)
	ValueFactory [▶ 1877]	The value factory (Inherited from DynamicValue [▶ 1869].)

Reference

DynamicPointerValue Class [▶ 1728]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.16.2 DynamicPointerValue Methods

The DynamicPointerValue [▶ 1728] type exposes the following members.

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1733]	Returns the enumeration of all dynamic member names. (Overrides DynamicValue.GetDynamicMemberNames. [▶ 1879].)
	Read [▶ 1879]	Reads the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	ReadAsync [▶ 1880]	read as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	ReadMember [▶ 1880]	Reads the specified member element. (Inherited from DynamicValue [▶ 1869].)
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	ToString [▶ 1882]	Returns a String that represents this instance. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue [▶ 1869].)
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value. (Inherited from DynamicValue [▶ 1869].)
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TryGetMemberValue [▶ 1733]	Tries the get member value. (Overrides DynamicValue.TryGetMemberValue(String, Object.) [▶ 1887].)
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue [▶ 1869].)
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue [▶ 1869].)
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays (Inherited from DynamicValue [▶ 1869].)
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value (Inherited from DynamicValue [▶ 1869].)
	Write [▶ 1893]	Writes the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	WriteAsync [▶ 1893]	write as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	WriteMember [▶ 1894]	Writes the specified member element. (Inherited from DynamicValue [▶ 1869].)

Reference

DynamicPointerValue Class [▶ 1728]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.16.2.1 DynamicPointerValue.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicPointerValue Class [▶ 1728]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.16.2.2 DynamicPointerValue.TryGetMemberValue Method

Tries the get member value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMemberValue(
    string name,
    out Object result
)
```

Parameters

name	Type: System.String The name.
result	Type: System.Object. The result.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IStructValue.TryGetMemberValue(String, Object.) [▶ 2169]

Exceptions

Exception	Condition
SymbolException [▶ 2401]	

Reference

DynamicPointerValue Class [▶ 1728]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17 DynamicReferenceInstance Class

Dynamic Reference Instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicReferenceInstance
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicReferenceInstance : DynamicSymbol,
    IReferenceInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicReferenceInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	ReferencedType [▶ 1740]	Gets the referenced type of the IReferenceInstance [▶ 2094]
	ResolvedByteSize [▶ 1740]	Gets the resolved byte size of the IReferenceInstance [▶ 2094].

	Name	Description
	ResolvedCategory [▶ 1741]	Gets the Category of the Referenced Symbol.
	ResolvedType [▶ 1741]	Gets the resolved type of the IReferenceInstance [▶ 2094].
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetIndex [▶ 1744]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1745]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Fields

	Name	Description
	normalizedDict [▶ 1746]	Dictionary of normalized Instance Names
	resolvedReferenceType [▶ 1747]	The resolved alias type
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.1 DynamicReferenceInstance Properties

The DynamicReferenceInstance [▶ 1734] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	ReferencedType [▶ 1740]	Gets the referenced type of the IReferenceInstance [▶ 2094]
	ResolvedByteSize [▶ 1740]	Gets the resolved byte size of the IReferenceInstance [▶ 2094].
	ResolvedCategory [▶ 1741]	Gets the Category of the Referenced Symbol.
	ResolvedType [▶ 1741]	Gets the resolved type of the IReferenceInstance [▶ 2094].
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.1.1 DynamicReferenceInstance.ReferencedType Property

Gets the referenced type of the IReferenceInstance [▶ 2094]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The referenced type

Implements

IReferenceInstance.ReferencedType [▶ 2097]

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.1.2 DynamicReferenceInstance.ResolvedByteSize Property

Gets the resolved byte size of the IReferenceInstance [▶ 2094].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ResolvedByteSize { get; }
```

Property Value

Type: Int32
The resolved byte size.

Implements

IReferenceInstance.ResolvedByteSize [▶ 2098]

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.1.3 DynamicReferenceInstance.ResolvedCategory Property

Gets the Category of the Referenced Symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory ResolvedCategory { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The resolved category.

Implements

IReferenceInstance.ResolvedCategory [▶ 2098]

Exceptions

Exception	Condition
NotImplementedException	

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.1.4 DynamicReferenceInstance.ResolvedType Property

Gets the resolved type of the IReferenceInstance [▶ 2094].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType ResolvedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The resolved type.

Implements

IReferenceInstance.ResolvedType [▶ 2099]

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.2 DynamicReferenceInstance Methods

The DynamicReferenceInstance [▶ 1734] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names. (Inherited from DynamicSymbol [▶ 1791].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetIndex [▶ 1744]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicSymbol [▶ 1791].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetIndex [▶ 1745]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.2.1 DynamicReferenceInstance.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.
result	Type: System.Object. The result of the index operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.2.2 DynamicReferenceInstance.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[][] is equal to 3.
value	Type: System.Object The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.3 DynamicReferenceInstance Events

The DynamicReferenceInstance [▶ 1734] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.4 DynamicReferenceInstance Fields

The DynamicReferenceInstance [▶ 1734] type exposes the following members.

Fields

	Name	Description
	normalizedDict [▶ 1746]	Dictionary of normalized Instance Names
	resolvedReferenceType [▶ 1747]	The resolved alias type
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.4.1 DynamicReferenceInstance.normalizedDict Field

Dictionary of normalized Instance Names

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Dictionary<string, ISymbol> normalizedDict
```

Field Value

Type: Dictionary.String, ISymbol [▶ 2176].

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.17.4.2 DynamicReferenceInstance.resolvedReferenceType Field

The resolved alias type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IDataType resolvedReferenceType
```

Field Value

Type: IDataType [▶ 1986]

Reference

DynamicReferenceInstance Class [▶ 1734]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.18 DynamicReferenceValue Class

Class DynamicReferenceValue.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicValue [▶ 1869]
    TwinCAT.TypeSystem.DynamicReferenceValue
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicReferenceValue : DynamicValue
```

The DynamicReferenceValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value) (Inherited from DynamicValue [▶ 1869].)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226] (Inherited from DynamicValue [▶ 1869].)
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from DynamicValue [▶ 1869].)
	ResolvedType [▶ 1875]	Gets the resolved type. (Inherited from DynamicValue [▶ 1869].)
	Symbol [▶ 1875]	Gets the symbol that is bound to this value. (Inherited from DynamicValue [▶ 1869].)
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue [▶ 1869].)
	UpdateMode [▶ 1876]	Gets / Sets the update mode (Inherited from DynamicValue [▶ 1869].)
	ValueFactory [▶ 1877]	The value factory (Inherited from DynamicValue [▶ 1869].)

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1879]	Returns the enumeration of all dynamic member names. (Inherited from DynamicValue [▶ 1869].)
	Read [▶ 1879]	Reads the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	ReadAsync [▶ 1880]	read as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	ReadMember [▶ 1751]	Reads the specified member element. (Overrides DynamicValue.ReadMember(ISymbol) [▶ 1880].)
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	ToString [▶ 1882]	Returns a String that represents this instance. (Inherited from DynamicValue [▶ 1869].)
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue [▶ 1869].)
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TryGetMemberValue [▶ 1887]	Tries the get member value. (Inherited from DynamicValue [▶ 1869].)
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue [▶ 1869].)
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue [▶ 1869].)
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays (Inherited from DynamicValue [▶ 1869].)
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value (Inherited from DynamicValue [▶ 1869].)
	Write [▶ 1893]	Writes the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	WriteAsync [▶ 1893]	write as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	WriteMember [▶ 1894]	Writes the specified member element. (Inherited from DynamicValue [▶ 1869].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DynamicValue [▶ 1869]

6.11.18.1 DynamicReferenceValue Properties

The DynamicReferenceValue [▶ 1747] type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value) (Inherited from DynamicValue [▶ 1869].)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226] (Inherited from DynamicValue [▶ 1869].)
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from DynamicValue [▶ 1869].)
	ResolvedType [▶ 1875]	Gets the resolved type. (Inherited from DynamicValue [▶ 1869].)
	Symbol [▶ 1875]	Gets the symbol that is bound to this value. (Inherited from DynamicValue [▶ 1869].)
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value. (Inherited from DynamicValue [▶ 1869].)
	UpdateMode [▶ 1876]	Gets / Sets the update mode (Inherited from DynamicValue [▶ 1869].)
	ValueFactory [▶ 1877]	The value factory (Inherited from DynamicValue [▶ 1869].)

Reference

DynamicReferenceValue Class [▶ 1747]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.18.2 DynamicReferenceValue Methods

The DynamicReferenceValue [▶ 1747] type exposes the following members.

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1879]	Returns the enumeration of all dynamic member names. (Inherited from DynamicValue [▶ 1869].)
	Read [▶ 1879]	Reads the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	ReadAsync [▶ 1880]	read as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	ReadMember [▶ 1751]	Reads the specified member element. (Overrides DynamicValue.ReadMember(ISymbol) [▶ 1880].)
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	ToString [▶ 1882]	Returns a String that represents this instance. (Inherited from DynamicValue [▶ 1869].)
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another. (Inherited from DynamicValue [▶ 1869].)
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations. (Inherited from DynamicValue [▶ 1869].)

	Name	Description
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element. (Inherited from DynamicValue [▶ 1869].)
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value. (Inherited from DynamicValue [▶ 1869].)
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TryGetMemberValue [▶ 1887]	Tries the get member value. (Inherited from DynamicValue [▶ 1869].)
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate. (Inherited from DynamicValue [▶ 1869].)
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method. (Inherited from DynamicValue [▶ 1869].)
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index. (Inherited from DynamicValue [▶ 1869].)
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays (Inherited from DynamicValue [▶ 1869].)
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicValue [▶ 1869].)
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value (Inherited from DynamicValue [▶ 1869].)
	Write [▶ 1893]	Writes the value (via ADS) (Inherited from DynamicValue [▶ 1869].)
	WriteAsync [▶ 1893]	write as an asynchronous operation. (Inherited from DynamicValue [▶ 1869].)
	WriteMember [▶ 1894]	Writes the specified member element. (Inherited from DynamicValue [▶ 1869].)

Reference

DynamicReferenceValue Class [▶ 1747]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.18.2.1 DynamicReferenceValue.ReadMember Method

Reads the specified member element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected override Object ReadMember(
    ISymbol memberInstance
)
```

Parameters

memberInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The member instance.

Return Value

Type: Object

Reference

DynamicReferenceValue Class [▶ 1747]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19 DynamicRpcStructInstance Class

Dynamic struct instance with RPC Methods.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicStructInstance [▶ 1779]
      TwinCAT.TypeSystem.DynamicRpcStructInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicRpcStructInstance : DynamicStructInstance,
    IRpcStructInstance, IStructInstance, ISymbol, IAttributedInstance, IInstance,
    IBitSize, IRpcCallableInstance
```

The DynamicRpcStructInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance [▶ 1779].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from DynamicStructInstance [▶ 1779].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	RpcMethods [▶ 1758]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1761]	Gets the dynamic member names. (Overrides DynamicStructInstance.GetDynamicMemberNames. [▶ 1789].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	InvokeRpcMethod(String, .Object.) [▶ 1762]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .Object..) [▶ 1763]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 1765]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 1767]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 1769]	Invokes the specified RPC Method asynchronously

	Name	Description
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1771]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicStructInstance.TryGetMember(GetMemberBinder, Object.) [▶ 1789].)
	TryInvokeMember [▶ 1772]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 1773]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 1775]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 1776]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1778]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Overrides DynamicStructInstance.TrySetMember(SetMemberBinder, Object) [▶ 1790].)

	Name	Description
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]

TwinCAT.TypeSystem.IStructInstance [▶ 2158]

TwinCAT.TypeSystem.IRpcStructInstance [▶ 2140]

TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104]

6.11.19.1 DynamicRpcStructInstance Properties

The DynamicRpcStructInstance [▶ 1752] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance [▶ 1779].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from DynamicStructInstance [▶ 1779].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	RpcMethods [▶ 1758]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.1.1 DynamicRpcStructInstance.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodCollection RpcMethods { get; }
```

Property Value

Type: IRpcMethodCollection [▶ 2127]
The methods.

Implements

IRpcCallableInstance.RpcMethods [▶ 2105]

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2 DynamicRpcStructInstance Methods

The DynamicRpcStructInstance [▶ 1752] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1761]	Gets the dynamic member names. (Overrides DynamicStructInstance.GetDynamicMemberNames. [▶ 1789].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	InvokeRpcMethod(String, .Object.) [▶ 1762]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .Object..) [▶ 1763]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 1765]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 1767]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 1769]	Invokes the specified RPC Method asynchronously
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1771]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicStructInstance.TryGetMember(GetMemberBinder, Object.) [▶ 1789].)
	TryInvokeMember [▶ 1772]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 1773]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 1775]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 1776]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1778]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Overrides DynamicStructInstance.TrySetMember(SetMemberBinder, Object) [▶ 1790].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.1 DynamicRpcStructInstance.GetDynamicMemberNames Method

Gets the dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
IEnumerable<System.String>.

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.2 DynamicRpcStructInstance.InvokeRpcMethod Method

Overload List

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 1762]	Invokes the specified RPC Method

	Name	Description
	InvokeRpcMethod(String, .Object., .Object..) [▶ 1763]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 1765]	Invokes the specified RPC Method

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.InvokeRpcMethod Method (String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL

Return Value

Type: Object
The return value of the Method (as object).

Implements

IRpcCallableInstance.InvokeRpcMethod(String, .Object.) [▶ 2107]

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

InvokeRpcMethod Overload [▶ 1761]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.InvokeRpcMethod Method (String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL
outParameters	Type: .System.Object.. The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IRpcCallableInstance.InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)
```

```
        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

InvokeRpcMethod Overload [▶ 1761]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.InvokeRpcMethod Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.

outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifiying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifiying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Implements

IRpcCallableInstance.InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
```

```
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

InvokeRpcMethod Overload [▶ 1761]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.3 DynamicRpcStructInstance.InvokeRpcMethodAsync Method

Overload List

	Name	Description
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 1767]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 1769]	Invokes the specified RPC Method asynchronously

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.InvokeRpcMethodAsync Method (String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethodAccess [▶ 2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess [▶ 2571] results contains the return value (ReturnValue [▶ 2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode [▶ 2559]) after the communication.

Implements

IRpcCallableInstance.InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program
```

```
        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

InvokeRpcMethodAsync Overload [▶ 1767]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.InvokeRpcMethodAsync Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.

retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethodAccess [▶ 2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess [▶ 2571] results contains the return value (ReturnValue [▶ 2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode [▶ 2559]) after the communication.

Implements

IRpcCallableInstance.InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
```

```
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

InvokeRpcMethodAsync Overload [▶ 1767]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.4 DynamicRpcStructInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.5 DynamicRpcStructInstance.TryInvokeMember Method

Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryInvokeMember(
    InvokeMemberBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.InvokeMemberBinder Provides information about the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleMethod". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
args	Type: .System.Object. The arguments that are passed to the object member during the invoke operation. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is derived from the DynamicObject class, args[][] is equal to 100.
result	Type: System.Object. The result of the member invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.6 DynamicRpcStructInstance.TryInvokeRpcMethod Method

Overload List

	Name	Description
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 1773]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .Object., Object.) [▶ 1775]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object., Object.) [▶ 1776]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.TryInvokeRpcMethod Method (String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: Int32
The return value of the Method (as object).

Implements

IRpcCallableInstance.TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
```

```
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

TryInvokeRpcMethod Overload [▶ 1773]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.TryInvokeRpcMethod Method (String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: Int32
The ADS Error Code.

Implements

IRpcCallableInstance.TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

TryInvokeRpcMethod Overload [▶ 1773]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicRpcStructInstance.TryInvokeRpcMethod Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String Name of the method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: Int32
AdsErrorCode.

Implements

IRpcCallableInstance.TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
```

```
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

DynamicRpcStructInstance Class [▶ 1752]

TryInvokeRpcMethod Overload [▶ 1773]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.2.7 DynamicRpcStructInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
value	Type: System.Object The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.19.3 DynamicRpcStructInstance Events

The DynamicRpcStructInstance [▶ 1752] type exposes the following members.

Events

	Name	Description
⚡	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
⚡	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicRpcStructInstance Class [▶ 1752]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20 DynamicStructInstance Class

Dynamic struct instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicStructInstance
      TwinCAT.TypeSystem.DynamicRpcStructInstance [▶ 1752]
      TwinCAT.TypeSystem.DynamicVirtualStructInstance [▶ 1894]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicStructInstance : DynamicSymbol,
    IStructInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicStructInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158].
	NormalizedDict [▶ 1786]	Dictionary of normalized Instance Names
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1789]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1789]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1790]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

	Name	Description
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.1 DynamicStructInstance Properties

The DynamicStructInstance [▶ 1779] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol) (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158].

	Name	Description
	NormalizedDict [▶ 1786]	Dictionary of normalized Instance Names
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.1.1 DynamicStructInstance.HasRpcMethods Property

Gets a value indicating whether this instance has RPC methods

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this instance has RPC methods; otherwise, false.

Implements

IStructInstance.HasRpcMethods [▶ 2161]

Remarks

If the struct instance supports RPC Methods, then the instance class is also supporting IRpcStructInstance [▶ 2140]

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IRpcStructInstance [▶ 2140]

TwinCAT.TypeSystem.IRpcMethod [▶ 2123]

TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133]

6.11.20.1.2 DynamicStructInstance.MemberInstances Property

Gets the member instances of the Struct Instance [▶ 2158].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolCollection<ISymbol> MemberInstances { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The member instances.

Implements

IStructInstance.MemberInstances [▶ 2162]

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.1.3 DynamicStructInstance.NormalizedDict Property

Dictionary of normalized Instance Names

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IDictionary<string, ISymbol> NormalizedDict { get; }
```

Property Value

Type: IDictionary.String, ISymbol [▶ 2176].

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.2 DynamicStructInstance Methods

The DynamicStructInstance [▶ 1779] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1789]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValue [▶ 1819]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath (Inherited from DynamicSymbol [▶ 1791].)
	OnTryReadValue [▶ 1821]	Handler function for the (Inherited from DynamicSymbol [▶ 1791].)
	OnTryWriteValue [▶ 1821]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValue [▶ 1823]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value. (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1789]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1790]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.2.1 DynamicStructInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.2.2 DynamicStructInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.2.3 DynamicStructInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
value	Type: System.Object The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.3 DynamicStructInstance Events

The DynamicStructInstance [▶ 1779] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.20.4 DynamicStructInstance Fields

The DynamicStructInstance [▶ 1779] type exposes the following members.

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicStructInstance Class [▶ 1779]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21 DynamicSymbol Class

Dynamic Symbol [▶ 2176] object.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol
    More... [▶ 1796]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicSymbol : DynamicObject,
    IDynamicSymbol, ISymbol, IAttributedInstance, IInstance, IBitSize,
    ISymbolFactoryServicesProvider, IValueSymbol, IValueRawSymbol, IHierarchicalSymbol, IValueAccess
orProvider,
    IContextMaskProvider
```

The DynamicSymbol type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol
	AccessRights [▶ 1798]	Gets the access rights.
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)
	Attributes [▶ 1799]	Gets the Symbol Attributes
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits.
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1801]	Gets the category.
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052]
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type.
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent.
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type.
	IsReadOnly [▶ 1808]	Indicates that this instance is read only.
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive.

	Name	Description
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static.
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),
	NotificationSettings [▶ 1810]	Gets the notification settings.
	Parent [▶ 1811]	Gets the parent Symbol
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes.
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176]
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].
	ValueEncoding [▶ 1813]	Gets the value encoding.

Methods

	Name	Description
	Equals [▶ 1816]	Equals
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names.
	GetHashCode [▶ 1817]	Gets the HashCode of the Address
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values.
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value.
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol.
	OnReadValue [▶ 1819]	Handler function for the
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols value asynchronously.
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath
	OnTryReadValue [▶ 1821]	Handler function for the
	OnTryWriteValue [▶ 1821]	Handler Function for writing value.
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value.
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol value.

	Name	Description
	OnWriteValue [▶ 1823]	Handler Function for writing value.
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value.
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation.
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol.
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol.
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously.
	ToString [▶ 1832]	Returns a String that represents this instance.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254]
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol.
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254]
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol.
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol.

	Name	Description
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254]

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Operators

	Name	Description
	Equality [▶ 1851]	Operator==
	Inequality [▶ 1851]	Implements the != operator.

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

Remarks

The DynamicSymbol adds dynamic run time behaviour to the ISymbol [▶ 2176]/IValueSymbol [▶ 2254]. That means e.g. for StructSymbols that .NET Properties are defined and dispatched at runtime to the structs fields like they are defined in TwinCAT / ADS Types. Indexed access to Array Symbols is another example where the dynamic runtime support takes place.

Examples

Sample for the dynamic resolution of Symbols:

Dynamic Symbol access

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
    {
    // Connect to the target device
    client.Connect(address);

    // Usage of "dynamic" Type and Symbols (>= .NET4 only)
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
    IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
```

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IDynamicSymbol [▶ 2004]

TwinCAT.TypeSystem.IValueSymbol [▶ 2254]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

System.Dynamic.DynamicObject

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol
    TwinCAT.TypeSystem.DynamicAliasInstance [▶ 1687]

TwinCAT.TypeSystem.DynamicArrayInstance [▶ 1698]
TwinCAT.TypeSystem.DynamicPointerInstance [▶ 1720]
TwinCAT.TypeSystem.DynamicReferenceInstance [▶ 1734]
TwinCAT.TypeSystem.DynamicStructInstance [▶ 1779]
TwinCAT.TypeSystem.DynamicUnionInstance [▶ 1860]

6.11.21.1 DynamicSymbol Properties

The DynamicSymbol [▶ 1791] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791]
	AccessRights [▶ 1798]	Gets the access rights.
	AllowIGIOAccess [▶ 1799]	Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)
	Attributes [▶ 1799]	Gets the Symbol Attributes
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits.
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes
	Category [▶ 1801]	Gets the category.
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052]
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791]
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type.
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent.
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type.
	IsReadOnly [▶ 1808]	Indicates that this instance is read only.

	Name	Description
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive.
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static.
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),
	NotificationSettings [▶ 1810]	Gets the notification settings.
	Parent [▶ 1811]	Gets the parent Symbol
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes.
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176]
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].
	ValueEncoding [▶ 1813]	Gets the value encoding.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.1 DynamicSymbol._InnerSymbol Property

Gets the inner symbol of this DynamicSymbol [▶ 1791]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IValueSymbol _InnerSymbol { get; }
```

Property Value

Type: IValueSymbol [▶ 2254]
The inner symbol.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.2 DynamicSymbol.AccessRights Property

Gets the access rights.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolAccessRights AccessRights { get; }
```

Property Value

Type: SymbolAccessRights [▶ 2396]
The access rights.

Implements

IValueSymbol.AccessRights [▶ 2259]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.3 DynamicSymbol.AllowIGIOAccess Property

Indicates, that the aggregates symbols is an IProcessImageAddress (and most probably IAdsSymbol)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected bool AllowIGIOAccess { get; }
```

Property Value

Type: Boolean

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.4 DynamicSymbol.Attributes Property

Gets the Symbol Attributes

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection [▶ 2211]
The attributes.

Implements

IAttributedInstance.Attributes [▶ 1982]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.5 DynamicSymbol.BitSize Property

Gets the size of the IDataType [▶ 1986] in bits.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int BitSize { get; }
```

Property Value

Type: Int32
The size of the bit.

Implements

IBitSize.BitSize [▶ 1984]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.6 DynamicSymbol.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ByteSize { get; }
```

Property Value

Type: Int32
The size of the byte.

Implements

IBitSize.ByteSize [▶ 1984]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.7 DynamicSymbol.Category Property

Gets the category.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCategory Category { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The category.

Implements

ISymbol.Category [▶ 2179]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.8 DynamicSymbol.Comment Property

Gets the comment of the IInstance [▶ 2052]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Comment { get; }
```

Property Value

Type: String
The comment.

Implements

IInstance.Comment [▶ 2053]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.9 DynamicSymbol.Connection Property

Gets the connection bound to this DynamicSymbol [▶ 1791]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]
The connection.

Implements

IValueSymbol.Connection [▶ 2260]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.10 DynamicSymbol.DataType Property

Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType DataType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Implements

IInstance.DataType [▶ 2054]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.11 DynamicSymbol.HasValue Property

Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool HasValue { get; }
```

Property Value

Type: Boolean
true if this instance has value; otherwise, false.

Implements

IValueRawSymbol.HasValue [▶ 2248]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.12 DynamicSymbol.InstanceName Property

Gets the name of the instance (without periods (.)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string InstanceName { get; }
```

Property Value

Type: String
The name of the instance.

Implements

IInstance.InstanceName [▶ 2054]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.13 DynamicSymbol.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.))

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Implements

IInstance.InstancePath [▶ 2055]

Remarks

If this path is relative or absolute depends on the context. IMember [▶ 2065] are using relative paths, ISymbol [▶ 2176]s are using absolute ones.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.14 DynamicSymbol.IsBitType Property

Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping; otherwise, false.

Implements

IBitSize.IsBitType [▶ 1984]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.15 DynamicSymbol.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Implements

IBitSize.IsByteAligned [▶ 1985]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.16 DynamicSymbol.IsContainerType Property

Gets a value indicating whether this Symbol is acontainer type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsContainerType { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Implements

ISymbol.IsContainerType [▶ 2179]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.17 DynamicSymbol.IsPersistent Property

Gets a value indicating whether this ISymbol [▶ 2176] is persistent.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPersistent { get; }
```

Property Value

Type: Boolean
true if this instance is persistent; otherwise, false.

Implements

ISymbol.IsPersistent [▶ 2180]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.18 DynamicSymbol.IsPointer Property

Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPointer { get; }
```

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Implements

IInstance.IsPointer [▶ 2055]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.19 DynamicSymbol.IsPrimitiveType Property

Gets a value indicating whether this instance is a primitive type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPrimitiveType { get; }
```

Property Value

Type: Boolean
true if this instance is primitive type; otherwise, false.

Implements

ISymbol.IsPrimitiveType [▶ 2180]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.20 DynamicSymbol.IsReadOnly Property

Indicates that this instance is read only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean

Implements

ISymbol.IsReadOnly [▶ 2181]

Remarks

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied [▶ 575] error.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.21 DynamicSymbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsRecursive { get; }
```

Property Value

Type: Boolean
true if this instance is recursive; otherwise, false.

Implements

ISymbol.IsRecursive [▶ 2181]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.22 DynamicSymbol.IsReference Property

Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReference { get; }
```

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Implements

IInstance.IsReference [▶ 2056]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.23 DynamicSymbol.IsStatic Property

Gets a value indicating whether this instance is static.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsStatic { get; }
```

Property Value

Type: Boolean
true if this instance is static; otherwise, false.

Implements

IInstance.IsStatic [▶ 2056]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.24 DynamicSymbol.NormalizedName Property

Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters),

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string NormalizedName { get; }
```

Property Value

Type: String
The normalized instance name (can be the same like InstanceName [▶ 2054]

Implements

IDynamicSymbol.NormalizedName [▶ 2007]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

IInstance.InstanceName [▶ 2054]

6.11.21.1.25 DynamicSymbol.NotificationSettings Property

Gets the notification settings.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public INotificationSettings NotificationSettings { get; set; }
```

Property Value

Type: INotificationSettings [▶ 972]
The notification settings.

Implements

IValueSymbol.NotificationSettings [▶ 2260]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.26 DynamicSymbol.Parent Property

Gets the parent Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Parent { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The parent.

Implements

ISymbol.Parent [▶ 2182]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.27 DynamicSymbol.Size Property

Gets the size of the IInstance [▶ 2052] in bytes.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Size { get; }
```

Property Value

Type: Int32
The size of the IInstance [▶ 2052] in bytes.

Implements

IBitSize.Size [▶ 1985]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.28 DynamicSymbol.SubSymbols Property

Gets the SubSymbols of the ISymbol [▶ 2176]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolCollection<ISymbol> SubSymbols { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].

Implements

ISymbol.SubSymbols [▶ 2182]

Remarks

Only used for Array and Struct instances. Otherwise empty

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.29 DynamicSymbol.TypeName Property

Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string TypeName { get; }
```

Property Value

Type: String
The name of the type.

Implements

IInstance.TypeName [▶ 2056]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.1.30 DynamicSymbol.ValueEncoding Property

Gets the value encoding.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Encoding ValueEncoding { get; }
```

Property Value

Type: Encoding
The value encoding.

Implements

IAttributedInstance.ValueEncoding [▶ 1982]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2 DynamicSymbol Methods

The DynamicSymbol [▶ 1791] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals
	GetDynamicMemberNames [▶ 1816]	Returns the enumeration of all dynamic member names.

	Name	Description
	GetHashCode [▶ 1817]	Gets the HashCode of the Address
	OnReadAnyValue [▶ 1817]	Handler function for reading ADS 'Any' Values.
	OnReadRawValue [▶ 1818]	Handler function for reading Raw symbol value.
	OnReadRawValueAsync [▶ 1818]	Handler function reading the raw value of the DynamicSymbol [▶ 1791].
	OnReadValue [▶ 1819]	Handler function for the
	OnReadValueAsync [▶ 1820]	Handler function reading the DynamicSymbols [▶ 1791] value asynchronously.
	OnSetInstanceName [▶ 1820]	Sets a new InstanceName InstancePath
	OnTryReadValue [▶ 1821]	Handler function for the
	OnTryWriteValue [▶ 1821]	Handler Function for writing value.
	OnWriteRawValue [▶ 1822]	Handler function for reading symbols raw value.
	OnWriteRawValueAsync [▶ 1823]	Handler function for writing the raw DynamicSymbol [▶ 1791] value.
	OnWriteValue [▶ 1823]	Handler Function for writing value.
	OnWriteValueAsync [▶ 1824]	Handler Function for writing value.
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation.
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791].
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791].
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously.
	ToString [▶ 1832]	Returns a String that represents this instance.
	TryGetMember [▶ 1833]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

	Name	Description
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254]
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791].
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254]
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791].
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791].
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254]

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.1 DynamicSymbol.Equals Method

Equals

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.2 DynamicSymbol.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.3 DynamicSymbol.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.4 DynamicSymbol.OnReadAnyValue Method

Handler function for reading ADS 'Any' Values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Object OnReadAnyValue(
    Type managedType
)
```

Parameters

managedType	Type: System.Type Managed type to read.

Return Value

Type: Object
System.Object.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.5 DynamicSymbol.OnReadRawValue Method

Handler function for reading Raw symbol value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual byte[] OnReadRawValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout.

Return Value

Type: .Byte.
System.Byte[].

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.6 DynamicSymbol.OnReadRawValueAsync Method

Handler function reading the raw value of the DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultReadRawAccess> OnReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].
Task<ResultReadRawAccess>.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.7 DynamicSymbol.OnReadValue Method

Handler function for the

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Object OnReadValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout.

Return Value

Type: Object
System.Object.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.8 DynamicSymbol.OnReadValueAsync Method

Handler function reading the DynamicSymbols [▶ 1791] value asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultReadValueAccess> OnReadValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
An asynchronous task returning the ResultReadValueAccess [▶ 2566] as result.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.9 DynamicSymbol.OnSetInstanceName Method

Sets a new InstanceName InstancePath

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnSetInstanceName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Instance name.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.10 DynamicSymbol.OnTryReadValue Method

Handler function for the

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual int OnTryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout	Type: System.Int32 The timeout.
value	Type: System.Object. The value.

Return Value

Type: Int32
The error Code.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.11 DynamicSymbol.OnTryWriteValue Method

Handler Function for writing value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual int OnTryWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout.

Return Value

Type: Int32

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.12 DynamicSymbol.OnWriteRawValue Method

Handler function for reading symbols raw value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnWriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.
timeout	Type: System.Int32 The timeout.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.13 DynamicSymbol.OnWriteRawValueAsync Method

Handler function for writing the raw DynamicSymbol [▶ 1791] value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultWriteAccess> OnWriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

Parameters

rawValue	Type: .System.Byte. The raw value to write.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
An asynchronous task object that represents the 'OnWriteRawValue' operation and returns a ResultWriteAccess [▶ 2575] as result.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.14 DynamicSymbol.OnWriteValue Method

Handler Function for writing value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void OnWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.15 DynamicSymbol.OnWriteValueAsync Method

Handler Function for writing value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Task<ResultWriteAccess> OnWriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
Task<WriteValueResult>.

Exceptions

Exception	Condition
CannotAccessVirtualSymbolException [▶ 1641]	

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.16 DynamicSymbol.ReadAnyValue Method

Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadAnyValue(
    Type managedType
)
```

Parameters

managedType	Type: System.Type The tp.	

Return Value

Type: Object
Read value (System.Object).

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

6.11.21.2.17 DynamicSymbol.ReadRawValue Method

Overload List

	Name	Description
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.ReadRawValue Method

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] ReadRawValue()
```

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue. [▶ 2250]

Reference

DynamicSymbol Class [▶ 1791]

ReadRawValue Overload [▶ 1825]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.ReadRawValue Method (Int32)

Reads the Symbols raw value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] ReadRawValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.

Field Value

Type: .Byte.
The raw value in bytes.

Return Value

Type: .Byte.
System.Byte[].

Implements

IValueRawSymbol.ReadRawValue(Int32) [▶ 2250]

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

DynamicSymbol Class [▶ 1791]

ReadRawValue Overload [▶ 1825]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.18 DynamicSymbol.ReadRawValueAsync Method

Read raw value as an asynchronous operation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Field Value

Type: Task.ResultReadRawAccess [▶ 2564].
The raw value.

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].
System.Byte[].

Implements

IValueRawSymbol.ReadRawValueAsync(CancellationToken) [▶ 2251]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.19 DynamicSymbol.ReadValue Method

Overload List

	Name	Description
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791].
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791].

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.ReadValue Method

Reads the value of this DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue()
```

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue. [▶ 2263]

Remarks

Examples

Dynamic Read access

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;

        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
```

```
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;


        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
        }
    }

    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
```

```
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
f"));
        }
    }
    }
}
```

Reference

DynamicSymbol Class [▶ 1791]

ReadValue Overload [▶ 1828]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.ReadValue Method (Int32)

Reads the value of this DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReadValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.	

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Implements

IValueSymbol.ReadValue(Int32) [▶ 2263]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

DynamicSymbol Class [▶ 1791]

ReadValue Overload [▶ 1828]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.20 DynamicSymbol.ReadValueAsync Method

Reads the Value of the IValueSymbol [▶ 2254] asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultReadValueAccess [▶ 2566] return value and contains the Value [▶ 2570] and the ErrorCode [▶ 2559].

Implements

IValueSymbol.ReadValueAsync(CancellationToken) [▶ 2264]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.21 DynamicSymbol.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.22 DynamicSymbol.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.23 DynamicSymbol.TryReadValue Method

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.
value	Type: System.Object. The symbol value.

Return Value

Type: Int32
The error code.

Implements

IValueSymbol.TryReadValue(Int32, Object.) [▶ 2265]

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.24 DynamicSymbol.TryWriteValue Method

Writes the specified value to the DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Return Value

Type: Int32
The error code.

Implements

IValueSymbol.TryWriteValue(Object, Int32) [▶ 2265]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Examples

Dynamic Write access

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
```

```
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;

        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;


        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);
```

```
        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
        }
    }

    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
f"));
        }
    }
    }
}
```

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.25 DynamicSymbol.UpdateAnyValue Method

Reads the value of this Value [▶ 2254] into the specified managed value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void UpdateAnyValue(
    ref Object valueObject
)
```

Parameters

valueObject	Type: System.Object. The managed object.

Return Value

Type:
Read value (System.Object).

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

6.11.21.2.26 DynamicSymbol.WriteAnyValue Method

Writes the value represented by the managed value to this Value [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteAnyValue(
    Object managedValue
)
```

Parameters

managedValue	Type: System.Object The managed value.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.ReadAnyValue(Type) [▶ 1825]

DynamicSymbol.UpdateAnyValue(Object.) [▶ 1837]

6.11.21.2.27 DynamicSymbol.WriteRawValue Method

Overload List

	Name	Description
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.WriteRawValue Method (.Byte.)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteRawValue(
    byte[] rawValue
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.

Implements

IValueRawSymbol.WriteRawValue(.Byte.) [▶ 2252]

Reference

DynamicSymbol Class [▶ 1791]

WriteRawValue Overload [▶ 1839]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.WriteRawValue Method (.Byte., Int32)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.
timeout	Type: System.Int32 The timeout.

Field Value

Type:
The value.

Implements

IValueRawSymbol.WriteRawValue(.Byte., Int32) [▶ 2252]

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

DynamicSymbol Class [▶ 1791]

WriteRawValue Overload [▶ 1839]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.28 DynamicSymbol.WriteRawValueAsync Method

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A task that represents the asynchronous read operation. The ResultRead [▶ 1008] parameter contains the total number of bytes read into the buffer (ReadBytes [▶ 1010]) and the ErrorCode [▶ 992] after execution..

Implements

IValueRawSymbol.WriteRawValueAsync(.Byte., CancellationToken) [▶ 2253]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.29 DynamicSymbol.WriteValue Method

Overload List

	Name	Description
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791].
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791].

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.WriteValue Method (Object)

Writes the specified value to the DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    Object value
)
```

Parameters

value	Type: System.Object The value.

Implements

IValueSymbol.WriteValue(Object) [▶ 2266]

Remarks

Examples

Dynamic Write access

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;

        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
```

```
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;


        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
        }
    }
```

```
    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
f"));
        }
    }
    }
}
```

Reference

DynamicSymbol Class [▶ 1791]

WriteValue Overload [▶ 1841]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicSymbol.WriteValue Method (Object, Int32)

Writes the specified value to the DynamicSymbol [▶ 1791].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void WriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Implements

IValueSymbol.WriteValue(Object, Int32) [▶ 2267]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Examples

Dynamic Write access

```
namespace Sample
{
    using System;
    using System.Diagnostics;
    using System.Threading;
    using TwinCAT;
    using TwinCAT.Ads;
    using TwinCAT.Ads.TypeSystem;
    using TwinCAT.Ads.ValueAccess;
    using TwinCAT.TypeSystem;
    using TwinCAT.TypeSystem.Generic;
    using TwinCAT.ValueAccess;

    class SymbolBrowserProgramV2DynamicTree
    {

    #region CODE_SAMPLE_SIMPLEDYNAMIC
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static async void Main(string[] args)
    {
        // Get the AdsAddress from command-line arguments
        AmsAddress address = ArgParser.Parse(args);

        CancellationTokenSource cancelSource = new CancellationTokenSource();
        CancellationToken cancel = cancelSource.Token;

        using (AdsClient client = new AdsClient())
        {
        // Connect to the target device
        client.Connect(address);

        // Usage of "dynamic" Type and Symbols (>= .NET4 only)
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        #endregion

        // Set the Default setting for Notifications
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 200,
 2000);

        // Get the Symbols (Dynamic Symbols)
        var resultSymbols = await ((IDynamicSymbolLoader)dynLoader).GetDynamicSymbolsAsync(cancel);

        dynamic dynamicSymbols = resultSymbols.Symbols;
        dynamic adsPort = dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.AdsPort;

        #region CODE_SAMPLE_SIMPLEDYNAMIC

        // Access Main Symbol with Dynamic Language Runtime support (DLR)
        // Dynamically created property "Main"
        //dynamic symMain = dynamicSymbols.Main;

        // Main is an 'VirtualSymbol' / Organizational unit that doesn't have a value
        // Calling ReadValue is not allowed
        //bool test = symMain.HasValue;
        //dynamic invalid = symMain.ReadValue();

        //Reading TaskInfo Value
        //
With calling ReadValueAsync() a 'snapshot' of the Symbols Instance is taken (reading async)
```

```
        ResultReadValueAccess resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo.
ReadValueAsync(cancel);
        dynamic vTaskInfoArray = resultRead.Value;

        // Getting the Snapshot time in UTC format
        DateTimeOffset timeStamp1 = vTaskInfoArray.TimeStamp;

        // Getting TaskInfo Symbol for Task 1
        dynamic symTaskInfo1 = dynamicSymbols.TwinCAT_SystemInfoVarList._TaskInfo[1];

        // Getting CycleCount Symbol
        dynamic symCycleCount = symTaskInfo1.CycleCount;


        // Take Snapshot value of the ApplicationInfo struct
        resultRead = await dynamicSymbols.TwinCAT_SystemInfoVarList._AppInfo.ReadValueAsync(cancel);
        dynamic vAppInfo = resultRead.Value;

        // Get the UTC Timestamp of the snapshot
        DateTimeOffset timeStamp2 = vAppInfo.TimeStamp;

        // Access the ProjectName of the ApplicationInfo Snapshot (type-safe!)
        string projectNameValue = vAppInfo.ProjectName;

        // Reading the CycleCount Value
        resultRead = await symTaskInfo1.CycleCount.ReadValueAsync(cancel);     // Taking a Value Sna
pshot
        int cycleCountValue = (int)resultRead.Value;
        #endregion

        // Registering for dynamic "ValueChanged" events for the Values
        // Using Default Notification settings
        symCycleCount.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueChange
d);

        // Override default notification settings
        symTaskInfo1.NotificationSettings = new NotificationSettings(AdsTransMode.Cyclic, 500, 0);

        // Register for ValueChanged event.
        symTaskInfo1.ValueChanged += new EventHandler<ValueChangedEventArgs>(taskInfo1Value_ValueCha
nged); // Struct Type

        Thread.Sleep(10000); // Sleep main thread for 10 Seconds
        }
        Console.WriteLine("CycleCount Changed events received: {0}",_cycleCountEvents);
        Console.WriteLine("taskInfo1 Changed events received: {0}", _taskInfo1Events);

        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    static object _notificationSynchronizer = new object();
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for the CycleCount ValueChanged event.
    /// </summary>
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock(_notificationSynchronizer)
        {
        Interlocked.Increment(ref _cycleCountEvents);
        // val is a type safe value of int!
        dynamic val = e.Value;
        uint intVal = val;

        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
        }
    }

    static int _taskInfo1Events = 0;

    /// <summary>
    /// Handler function for the TaskInfo ValueChanged event.
    /// </summary>
```

```
    /// <param name="sender">The sender.</param>
    /// <param name="e">The e.</param>
    static void taskInfo1Value_ValueChanged(object sender, ValueChangedEventArgs e)
    {
        lock (_notificationSynchronizer)
        {
        Interlocked.Increment(ref _taskInfo1Events);
        dynamic val = e.Value;
        DateTimeOffset changedTime = e.DateTime.ToLocalTime(); // Convert to local time

        // Val is a during Runtime created struct type and contains
        // the same Properties as related PLC object.
        int cycleTime = val.CycleTime;
        Console.WriteLine("TaskInfo1Value changed TimeStamp: {0}", changedTime.ToString("HH:mm:ss:ff
f"));
        }
    }
    }
}
```

Reference

DynamicSymbol Class [▶ 1791]

WriteValue Overload [▶ 1841]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.2.30 DynamicSymbol.WriteValueAsync Method

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultWriteAccess [▶ 2575] return value and contains the ErrorCode [▶ 2559].

Implements

IValueSymbol.WriteValueAsync(Object, CancellationToken) [▶ 2268]

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.3 DynamicSymbol Events

The DynamicSymbol [▶ 1791] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.3.1 DynamicSymbol.RawValueChanged Event

Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler.RawValueChangedEventArgs [▶ 2289].

Implements

IValueRawSymbol.RawValueChanged [▶ 2254]

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.3.2 DynamicSymbol.ValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

Value

Type: System.EventHandler.ValueChangedEventArgs [▶ 2439].

Implements

IValueSymbol.ValueChanged [▶ 2269]

Remarks

Examples

Use Dynamic Notifications

```
using System;
using System.Diagnostics;
using System.Threading;
using TwinCAT;
using TwinCAT.Ads;
using TwinCAT.Ads.TypeSystem;
using TwinCAT.Ads.ValueAccess;
using TwinCAT.TypeSystem;
using TwinCAT.TypeSystem.Generic;

namespace Sample
{
    class SymbolBrowserV2Notifications
    {
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
        // Parse the Command Line Parameters.
        AmsAddress address = ArgParser.Parse(args);

        #region DEFAULTNOTIFICATON_SAMPLE

        // Create AdsClient object
        using (AdsClient client = new AdsClient())
        {
        // No automatic Synchronization (necessary for Console applications without message loop)
        //client.Synchronize = false;

        // Connect to client
        client.Connect(address);

        // Usage of 'dynamic' type/symbol loader
        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree, ValueA
ccessMode.IndexGroupOffsetPreferred);
```

```
        IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);

        // Set the DefaultNotification Properties
        dynLoader.DefaultNotificationSettings = new NotificationSettings(AdsTransMode.ClientOnChange
, 200, 2000);

        // Determine the symbols
        dynamic dynamicSymbols = ((IDynamicSymbolLoader)dynLoader).SymbolsDynamic;

        // Task 1 Symbol (build in symbol)
        dynamic task1Symbol = dynamicSymbols._TaskInfo[1];

        // CycleCount Symbol
        dynamic cycleCountSymbol = task1Symbol.CycleCount;

        // Override Notification Setting for Cycle Count Symbol
        cycleCountSymbol.NotificationSettings = new NotificationSettings(AdsTransMode.OnChange, 250,
 0);

        // Register Dynamic Value Changed event.
        cycleCountSymbol.ValueChanged += new EventHandler<ValueChangedEventArgs>(cycleCount_ValueCha
nged);
#endregion
        // Sleep main thread to receive notifications
        Thread.Sleep(10000);
#region DEFAULTNOTIFICATON_SAMPLE
        }
#endregion
        Console.WriteLine("CycleCount Changed events received: {0}", _cycleCountEvents);
        Console.WriteLine("");
        Console.WriteLine("Press [Enter] for leave:");
        Console.ReadLine();
    }

    /// <summary>
    /// The cycle count event counter
    /// </summary>
    static int _cycleCountEvents = 0;

    /// <summary>
    /// Handler function for CycleCount changed events.
    /// </summary>
    /// <param name="sender">Event sender.</param>
    /// <param name="args">Event arguments.</param>
    static void cycleCount_ValueChanged(object sender, ValueChangedEventArgs args)
    {
        Interlocked.Increment(ref _cycleCountEvents);

        // Use Value as dynamic (type safe: INT)  object.
        dynamic val = args.Value;
        int intVal = val;

        DateTimeOffset changedTime = args.DateTime.ToLocalTime(); // Convert UTC to local time
        Console.WriteLine("CycleCount changed to: {0}, TimeStamp: {1}", intVal, changedTime.ToString
("HH:mm:ss:fff"));
    }
    }
}
```

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.4 DynamicSymbol Operators

The DynamicSymbol [▶ 1791] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 1851]	Operator==
	Inequality [▶ 1851]	Implements the != operator.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.4.1 DynamicSymbol.Equality Operator

Operator==

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    DynamicSymbol o1,
    DynamicSymbol o2
)
```

Parameters

o1	Type: TwinCAT.TypeSystem.DynamicSymbol [▶ 1791] The o1.
o2	Type: TwinCAT.TypeSystem.DynamicSymbol [▶ 1791] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.4.2 DynamicSymbol.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    DynamicSymbol o1,
    DynamicSymbol o2
)
```

Parameters

o1	Type: TwinCAT.TypeSystem.DynamicSymbol [▶ 1791] The o1.
o2	Type: TwinCAT.TypeSystem.DynamicSymbol [▶ 1791] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.5 DynamicSymbol Fields

The DynamicSymbol [▶ 1791] type exposes the following members.

Fields

	Name	Description
	syncObject [▶ 1852]	Synchronization object

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.21.5.1 DynamicSymbol.syncObject Field

Synchronization object

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected Object syncObject
```

Field Value

Type: Object

Reference

DynamicSymbol Class [▶ 1791]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22 DynamicSymbolsCollection Class

Dynamic (Expandable) Symbols collection.

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbolsCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicSymbolsCollection : DynamicObject,
    IDynamicSymbolsCollection, IDynamicMetaObjectProvider, IEnumerable<ISymbol>, IEnumerable
```

The DynamicSymbolsCollection type exposes the following members.

Constructors

	Name	Description
	DynamicSymbolsCollection(IEnumerable.ISymbol.) [▶ 1854]	Initializes a new instance of the DynamicSymbolsCollection class (for internal use only)
	DynamicSymbolsCollection(SymbolCollection.ISymbol.) [▶ 1855]	Initializes a new instance of the DynamicSymbolsCollection class (for internal use only)

Properties

	Name	Description
	Item [▶ 1856]	Gets the DynamicSymbol [▶ 1791] with the specified name.

Methods

	Name	Description
S	Empty [▶ 1857]	Gets an empty collection.

	Name	Description
	GetDynamicMemberNames [▶ 1857]	Returns the enumeration of all dynamic member names.
	GetEnumerator [▶ 1858]	Gets the enumerator.
	TryGetInstance [▶ 1858]	Tries to get the IInstance [▶ 2052]. of the specified path.
	TryGetMember [▶ 1859]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Remarks

The DynamicSymbolsCollection collection adds dynamically its child Symbols as Members (for access like "Main.Symbol")

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.1 DynamicSymbolsCollection Constructor

Overload List

	Name	Description
	DynamicSymbolsCollection(IEnumerable.ISymbol.) [▶ 1854]	Initializes a new instance of the DynamicSymbolsCollection [▶ 1853] class (for internal use only)
	DynamicSymbolsCollection(SymbolCollection.ISymbol.) [▶ 1855]	Initializes a new instance of the DynamicSymbolsCollection [▶ 1853] class (for internal use only)

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.1.1 DynamicSymbolsCollection Constructor (IEnumerable.ISymbol.)

Initializes a new instance of the DynamicSymbolsCollection [▶ 1853] class (for internal use only)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DynamicSymbolsCollection(
    IEnumerable<ISymbol> symbols
)
```

Parameters

symbols	Type: System.Collections.Generic.IEnumerable.ISymbol [▶ 2176]. The symbols.

Exceptions

Exception	Condition
ArgumentNullException	symbols

Reference

DynamicSymbolsCollection Class [▶ 1853]

DynamicSymbolsCollection Overload [▶ 1854]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.1.2 DynamicSymbolsCollection Constructor (SymbolCollection.ISymbol.)

Initializes a new instance of the DynamicSymbolsCollection [▶ 1853] class (for internal use only)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DynamicSymbolsCollection(
    SymbolCollection<ISymbol> symbols
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.Generic.SymbolCollection [▶ 2526].ISymbol [▶ 2176]. The symbols.

Exceptions

Exception	Condition
ArgumentNullException	symbols

Reference

DynamicSymbolsCollection Class [▶ 1853]

DynamicSymbolsCollection Overload [▶ 1854]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.2 DynamicSymbolsCollection Properties

The DynamicSymbolsCollection [▶ 1853] type exposes the following members.

Properties

	Name	Description
	Item [▶ 1856]	Gets the DynamicSymbol [▶ 1791] with the specified name.

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.2.1 DynamicSymbolsCollection.Item Property

Gets the DynamicSymbol [▶ 1791] with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DynamicSymbol this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: DynamicSymbol [▶ 1791]
DynamicSymbol.

Exceptions

Exception	Condition
KeyNotFoundException	Symbol name not found in DynamicSymbols collection!

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3 DynamicSymbolsCollection Methods

The DynamicSymbolsCollection [▶ 1853] type exposes the following members.

Methods

	Name	Description
	Empty [▶ 1857]	Gets an empty collection.
	GetDynamicMemberNames [▶ 1857]	Returns the enumeration of all dynamic member names.
	GetEnumerator [▶ 1858]	Gets the enumerator.
	TryGetInstance [▶ 1858]	Tries to get the IInstance [▶ 2052]. of the specified path.
	TryGetMember [▶ 1859]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3.1 DynamicSymbolsCollection.Empty Method

Gets an empty collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static DynamicSymbolsCollection Empty()
```

Return Value

Type: DynamicSymbolsCollection [▶ 1853]
DynamicSymbolsCollection.

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3.2 DynamicSymbolsCollection.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3.3 DynamicSymbolsCollection.GetEnumerator Method

Gets the enumerator.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<ISymbol> GetEnumerator()
```

Return Value

Type: IEnumerator.ISymbol [▶ 2176].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3.4 DynamicSymbolsCollection.TryGetInstance Method

Tries to get the IInstance [▶ 2052]. of the specified path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInstance(
    string instanceSpecifier,
    out ISymbol symbol
)
```

Parameters

instanceSpecifier	Type: System.String The instance path or Instance Name (dependent of Mode [▶ 2468] setting)
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]. The symbol.

Return Value

Type: Boolean
true if the IInstance [▶ 2052] is found; otherwise, false

Exceptions

Exception	Condition
ArgumentNullException	instancePath
ArgumentException	

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.22.3.5 DynamicSymbolsCollection.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicSymbolsCollection Class [▶ 1853]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23 DynamicUnionInstance Class

Dynamic union instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicUnionInstance
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicUnionInstance : DynamicSymbol,
    IUnionInstance, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The DynamicUnionInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	FieldInstances [▶ 1865]	Gets the member instances of the Struct Instance [▶ 2158].
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1867]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1867]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1868]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

	Name	Description
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.1 DynamicUnionInstance Properties

The DynamicUnionInstance [▶ 1860] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	FieldInstances [▶ 1865]	Gets the member instances of the Struct Instance [▶ 2158].
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.1.1 DynamicUnionInstance.FieldInstances Property

Gets the member instances of the Struct Instance [▶ 2158].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbolCollection<ISymbol> FieldInstances { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The member instances.

Implements

IUnionInstance.FieldInstances [▶ 2223]

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.2 DynamicUnionInstance Methods

The DynamicUnionInstance [▶ 1860] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	GetDynamicMemberNames [▶ 1867]	Returns the enumeration of all dynamic member names. (Overrides DynamicSymbol.GetDynamicMemberNames. [▶ 1816].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1867]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Overrides DynamicSymbol.TryGetMember(GetMemberBinder, Object.) [▶ 1833].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1868]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.2.1 DynamicUnionInstance.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.2.2 DynamicUnionInstance.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.2.3 DynamicUnionInstance.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
value	Type: System.Object The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.23.3 DynamicUnionInstance Events

The DynamicUnionInstance [▶ 1860] type exposes the following members.

Events

	Name	Description
⚡	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
⚡	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicUnionInstance Class [▶ 1860]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24 DynamicValue Class

Dynamic value (uses RuntimeBinding for ISymbol [▶ 2176] value reading / writing).

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicValue
    TwinCAT.TypeSystem.DynamicPointerValue [▶ 1728]
    TwinCAT.TypeSystem.DynamicReferenceValue [▶ 1747]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DynamicValue : DynamicObject,
    IDynamicValue, IDynamicMetaObjectProvider, IValue, IStructValue, IArrayValue
```

The DynamicValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data.
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226]
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value.
	ResolvedType [▶ 1875]	Gets the resolved type.
	Symbol [▶ 1875]	Gets the symbol that is bound to this value.
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value.
	UpdateMode [▶ 1876]	Gets / Sets the update mode
	ValueFactory [▶ 1877]	The value factory

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1879]	Returns the enumeration of all dynamic member names.
	Read [▶ 1879]	Reads the value (via ADS)
	ReadAsync [▶ 1880]	read as an asynchronous operation.
	ReadMember [▶ 1880]	Reads the specified member element.
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value.
	ToString [▶ 1882]	Returns a String that represents this instance.
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another.
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values.
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element.
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value.

	Name	Description
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.
	TryGetMemberValue [▶ 1887]	Tries the get member value.
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value.
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value
	Write [▶ 1893]	Writes the value (via ADS)
	WriteAsync [▶ 1893]	write as an asynchronous operation.
	WriteMember [▶ 1894]	Writes the specified member element.

Remarks

The DynamicValue adds dynamic run time behaviour to the IValue [▶ 2226]Value/IValue [▶ 2226]. That means e.g. for struct values that .NET Properties are on-the-fly defined and dispatched at runtime just like defined in the structs structs data type definition. Another example is the access of Array Element values through indexes.

Examples

Sample for the dynamic resolution of Symbols and reading values:

Dynamic Symbol access

```
/// <summary>
/// Defines the entry point of the application.
/// </summary>
/// <param name="args">The arguments.</param>
static async void Main(string[] args)
{
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    CancellationTokenSource cancelSource = new CancellationTokenSource();
    CancellationToken cancel = cancelSource.Token;

    using (AdsClient client = new AdsClient())
```

```
    {
    // Connect to the target device
    client.Connect(address);

    // Usage of "dynamic" Type and Symbols (>= .NET4 only)
    SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.DynamicTree);
    IAdsSymbolLoader dynLoader = (IAdsSymbolLoader)SymbolLoaderFactory.Create(client, settings);
```

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]

System.Dynamic.DynamicObject

TwinCAT.TypeSystem.IValue [▶ 2226]

6.11.24.1 DynamicValue Properties

The DynamicValue [▶ 1869] type exposes the following members.

Properties

	Name	Description
	Age [▶ 1872]	Gets the age of the value (last successful read of the value)
	CachedRaw [▶ 1873]	Gets the cached Raw internal Data.
	DataType [▶ 1874]	Gets the data type bound to this IValue [▶ 2226]
	IsPrimitive [▶ 1874]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value.
	ResolvedType [▶ 1875]	Gets the resolved type.
	Symbol [▶ 1875]	Gets the symbol that is bound to this value.
	TimeStamp [▶ 1876]	Gets the Time stamp of the last successful read of the Value.
	UpdateMode [▶ 1876]	Gets / Sets the update mode
	ValueFactory [▶ 1877]	The value factory

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.1 DynamicValue.Age Property

Gets the age of the value (last successful read of the value)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TimeSpan Age { get; }
```

Property Value

Type: TimeSpan
The age.

Implements

IValue.Age [▶ 2228]

Exceptions

Exception	Condition
NotImplementedException	

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicValue.TimeStamp [▶ 1876]

6.11.24.1.2 DynamicValue.CachedRaw Property

Gets the cached Raw internal Data.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] CachedRaw { get; }
```

Property Value

Type: .Byte.
The raw cached data.

Implements

IValue.CachedRaw [▶ 2228]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.3 DynamicValue.DataType Property

Gets the data type bound to this IValue [▶ 2226]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataType DataType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Implements

IValue.DataType [▶ 2228]

Exceptions

Exception	Condition
NotImplementedException	

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.4 DynamicValue.IsPrimitive Property

Gets a value indicating whether this IValue [▶ 2226] is a primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Implements

IValue.IsPrimitive [▶ 2229]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.5 DynamicValue.ResolvedType Property

Gets the resolved type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IDataType ResolvedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
Resolved type.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.6 DynamicValue.Symbol Property

Gets the symbol that is bound to this value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Symbol { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The symbol.

Implements

IValue.Symbol [▶ 2229]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.7 DynamicValue.TimeStamp Property

Gets the Time stamp of the last successful read of the Value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The read time stamp.

Implements

IValue.TimeStamp [▶ 2230]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.1.8 DynamicValue.UpdateMode Property

Gets / Sets the update mode

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ValueUpdateMode UpdateMode { get; set; }
```

Property Value

Type: ValueUpdateMode [▶ 150]
The update mode.

Implements

IValue.UpdateMode [▶ 2230]

Remarks

The default value is initialized by the creating Value Factory.

Mode	Description
Immediately [▶ 150]	Writes the values of this DynamicValue [▶ 1869]"/> instantly when setting its value or the value of its child members/elements.
Triggered [▶ 150]	Caches internally the value of this DynamicValue [▶ 1869]"/> until the Write. [▶ 1893] method is called. This reduces ADS rountrips, if one or more member/ element values should be changed. Furthermore the write on the destination system happens consistently in one ADS Write operation, which could be important for dependent properties/members/elements.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.ValueUpdateMode [▶ 150]

6.11.24.1.9 DynamicValue.ValueFactory Property

The value factory

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IAccessorValueFactory ValueFactory { get; }
```

Property Value

Type: IAccessorValueFactory [▶ 2552]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2 DynamicValue Methods

The DynamicValue [▶ 1869] type exposes the following members.

Methods

	Name	Description
	GetDynamicMemberNames [▶ 1879]	Returns the enumeration of all dynamic member names.
	Read [▶ 1879]	Reads the value (via ADS)
	ReadAsync [▶ 1880]	read as an asynchronous operation.

	Name	Description
	ReadMember [▶ 1880]	Reads the specified member element.
	ResolveValue [▶ 1881]	Resolves the Value object to its primitive value.
	ToString [▶ 1882]	Returns a String that represents this instance.
	TryConvert [▶ 1882]	Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another.
	TryGetArrayElementValues [▶ 1883]	Returns Array Element values.
	TryGetIndex [▶ 1883]	Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element.
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value.
	TryGetMember [▶ 1886]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.
	TryGetMemberValue [▶ 1887]	Tries the get member value.
	TryInvoke [▶ 1887]	Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.
	TryInvokeMember [▶ 1888]	Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.
	TryResolveValue [▶ 1889]	Tries to resolves the Value object to its primitive value.
	TrySetIndex [▶ 1890]	Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.
	TrySetIndexValue [▶ 1890]	Tries to set the indexed value on Arrays
	TrySetMember [▶ 1891]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.
	TrySetMemberValue [▶ 1892]	Tries to Set a Member/Property Value
	Write [▶ 1893]	Writes the value (via ADS)
	WriteAsync [▶ 1893]	write as an asynchronous operation.

	Name	Description
	WriteMember [▶ 1894]	Writes the specified member element.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.1 DynamicValue.GetDynamicMemberNames Method

Returns the enumeration of all dynamic member names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual IEnumerable<string> GetDynamicMemberNames()
```

Return Value

Type: IEnumerable.String.
A sequence that contains dynamic member names.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.2 DynamicValue.Read Method

Reads the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Read()
```

Implements

IValue.Read. [▶ 2231]

Exceptions

Exception	Condition
SymbolException [▶ 2401]	

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.3 DynamicValue.ReadAsync Method

read as an asynchronous operation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultAccess> ReadAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAccess [▶ 2556].
Task<ReadValueResult>.

Implements

IValue.ReadAsync(CancellationToken) [▶ 2231]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.4 DynamicValue.ReadMember Method

Reads the specified member element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual Object ReadMember(
    ISymbol memberInstance
)
```

Parameters

memberInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The member instance.

Return Value

Type: Object

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.5 DynamicValue.ResolveValue Method

Resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ResolveValue(
    bool resolveEnumToPrimitive
)
```

Parameters

resolveEnumToPrimitive	Type: System.Boolean if set to trueIEnumValue [▶ 2028]s are resolved to their primitives also.

Return Value

Type: Object
System.Object.

Implements

IValue.ResolveValue(Boolean) [▶ 2232]

Remarks

If the value is not primitive, this method returns the IValue [▶ 2226] itself.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.6 DynamicValue.ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.7 DynamicValue.TryConvert Method

Provides implementation for type conversion operations. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that convert an object from one type to another.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryConvert(
    ConvertBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.ConvertBinder Provides information about the conversion operation. The binder.Type property provides the type to which the object must be converted. For example, for the statement (String)sampleObject in C# (CType(sampleObject, Type) in Visual Basic), where sampleObject is an instance of the class derived from the DynamicObject class, binder.Type returns the String type. The binder.Explicit property provides information about the kind of conversion that occurs. It returns true for explicit conversion and false for implicit conversion.
result	Type: System.Object. The result of the type conversion operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.8 DynamicValue.TryGetArrayElementValues Method

Returns Array Element values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetArrayElementValues(
    out IEnumerable<Object> elementValues
)
```

Parameters

elementValues	Type: System.Collections.Generic.IEnumerable.Object.. The element values.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IArrayValue.TryGetArrayElementValues(IEnumerable.Object..) [▶ 1978]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.9 DynamicValue.TryGetIndex Method

Provides the implementation for operations that get a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for indexing operations.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetIndex(
    GetIndexBinder binder,
    Object[] indexes,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] operation in C# (sampleObject(3) in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.
result	Type: System.Object. The result of the index operation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.10 DynamicValue.TryGetIndexValue Method

Overload List

	Name	Description
	TryGetIndexValue(.Int32., Object.) [▶ 1884]	Reads the specified array element.
	TryGetIndexValue(.Object., Object.) [▶ 1885]	Tries the get index value.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicValue.TryGetIndexValue Method (.Int32., Object.)

Reads the specified array element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetIndexValue(
    int[] indices,
    out Object value
)
```

Parameters

indices	Type: .System.Int32. The indices.
value	Type: System.Object. The value.

Return Value

Type: Boolean
System.Object.

Implements

IArrayValue.TryGetIndexValue(.Int32., Object.) [▶ 1978]

Reference

DynamicValue Class [▶ 1869]

TryGetIndexValue Overload [▶ 1884]

TwinCAT.TypeSystem Namespace [▶ 1622]

DynamicValue.TryGetIndexValue Method (.Object., Object.)

Tries the get index value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetIndexValue(
    Object[] indexes,
    out Object result
)
```

Parameters

indexes	Type: .System.Object. The indexes.
result	Type: System.Object. The result.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

DynamicValue Class [▶ 1869]

TryGetIndexValue Overload [▶ 1884]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.11 DynamicValue.TryGetMember Method

Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetMember(
    GetMemberBinder binder,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.GetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the Console.WriteLine(sampleObject.SampleProperty) statement, where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.
result	Type: System.Object. The result of the get operation. For example, if the method is called for a property, you can assign the property value to result.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a run-time exception is thrown.)

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.12 DynamicValue.TryGetMemberValue Method

Tries the get member value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetMemberValue(
    string name,
    out Object result
)
```

Parameters

name	Type: System.String The name.
result	Type: System.Object. The result.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IStructValue.TryGetMemberValue(String, Object.) [▶ 2169]

Exceptions

Exception	Condition
SymbolException [▶ 2401]	

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.13 DynamicValue.TryInvoke Method

Provides the implementation for operations that invoke an object. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as invoking an object or a delegate.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryInvoke(
    InvokeBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.InvokeBinder Provides information about the invoke operation.
args	Type: .System.Object. The arguments that are passed to the object during the invoke operation. For example, for the sampleObject(100) operation, where sampleObject is derived from the DynamicObject class, args is equal to 100.
result	Type: System.Object. The result of the object invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.14 DynamicValue.TryInvokeMember Method

Provides the implementation for operations that invoke a member. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as calling a method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryInvokeMember(
    InvokeMemberBinder binder,
    Object[] args,
    out Object result
)
```

Parameters

binder	Type: System.Dynamic.InvokeMemberBinder Provides information about the dynamic operation. The binder.Name property provides the name of the member on which the dynamic operation is performed. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleMethod". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

args	Type: .System.Object. The arguments that are passed to the object member during the invoke operation. For example, for the statement sampleObject.SampleMethod(100), where sampleObject is derived from the DynamicObject class, args[0] is equal to 100.
result	Type: System.Object. The result of the member invocation.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.15 DynamicValue.TryResolveValue Method

Tries to resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryResolveValue(
    bool resolveEnumToPrimitive,
    out Object value
)
```

Parameters

resolveEnumToPrimitive	Type: System.Boolean if set to trueIEnumValue [▶ 2028]s are resolved to their primitives also.
value	Type: System.Object. The value.

Return Value

Type: Boolean
true if value can be resolved, false otherwise.

Implements

IValue.TryResolveValue(Boolean, Object.) [▶ 2232]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.16 DynamicValue.TrySetIndex Method

Provides the implementation for operations that set a value by index. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations that access objects by a specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetIndex(
    SetIndexBinder binder,
    Object[] indexes,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetIndexBinder Provides information about the operation.
indexes	Type: .System.Object. The indexes that are used in the operation. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, indexes[0] is equal to 3.
value	Type: System.Object The value to set to the object that has the specified index. For example, for the sampleObject[3] = 10 operation in C# (sampleObject(3) = 10 in Visual Basic), where sampleObject is derived from the DynamicObject class, value is equal to 10.

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.17 DynamicValue.TrySetIndexValue Method

Tries to set the indexed value on Arrays

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TrySetIndexValue(
    Object[] indexes,
    Object value
)
```

Parameters

indexes	Type: .System.Object. The indexes.
value	Type: System.Object The value.

Return Value

Type: Boolean
true if succeeded, false otherwise.

Implements

IArrayValue.TrySetIndexValue(.Object., Object) [▶ 1979]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.18 DynamicValue.TrySetMember Method

Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TrySetMember(
    SetMemberBinder binder,
    Object value
)
```

Parameters

binder	Type: System.Dynamic.SetMemberBinder Provides information about the object that called the dynamic operation. The binder.Name property provides the name of the member to which the value is being assigned. For example, for the statement sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, binder.Name returns "SampleProperty". The binder.IgnoreCase property specifies whether the member name is case-sensitive.

value	Type: System.Object The value to set to the member. For example, for sampleObject.SampleProperty = "Test", where sampleObject is an instance of the class derived from the DynamicObject class, the value is "Test".

Return Value

Type: Boolean
true if the operation is successful; otherwise, false. If this method returns false, the run-time binder of the language determines the behavior. (In most cases, a language-specific run-time exception is thrown.)

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.19 DynamicValue.TrySetMemberValue Method

Tries to Set a Member/Property Value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TrySetMemberValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String The name of the member
value	Type: System.Object The value.

Return Value

Type: Boolean
true if succeeded, otherwise false otherwise.

Implements

IStructValue.TrySetMemberValue(String, Object) [▶ 2170]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.20 DynamicValue.Write Method

Writes the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Write()
```

Implements

IValue.Write. [▶ 2233]

Exceptions

Exception	Condition
SymbolException [▶ 2401]	

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.21 DynamicValue.WriteAsync Method

write as an asynchronous operation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Task<ResultWriteAccess> WriteAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
Task<WriteValueResult>.

Implements

IValue.WriteAsync(CancellationToken) [▶ 2233]

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.24.2.22 DynamicValue.WriteMember Method

Writes the specified member element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected virtual void WriteMember(
    ISymbol memberInstance,
    Object value
)
```

Parameters

memberInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The member instance.
value	Type: System.Object The value.

Reference

DynamicValue Class [▶ 1869]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.25 DynamicVirtualStructInstance Class

Dynamic struct instance

Inheritance Hierarchy

System.Dynamic.DynamicObject
  TwinCAT.TypeSystem.DynamicSymbol [▶ 1791]
    TwinCAT.TypeSystem.DynamicStructInstance [▶ 1779]
      TwinCAT.TypeSystem.DynamicVirtualStructInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public sealed class DynamicVirtualStructInstance : DynamicStructInstance,
    IVirtualStructInstance, IStructInstance, ISymbol, IAttributedInstance, IInstance,
    IBitSize
```

The DynamicVirtualStructInstance type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance [▶ 1779].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from DynamicStructInstance [▶ 1779].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Methods

	Name	Description
	AddMember [▶ 1900]	Adds an member instance.
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1789]	Returns the enumeration of all dynamic member names. (Inherited from DynamicStructInstance [▶ 1779].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	TryGetMember [▶ 1789]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicStructInstance [▶ 1779].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1790]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicStructInstance [▶ 1779].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Events

	Name	Description
	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.25.1 DynamicVirtualStructInstance Properties

The DynamicVirtualStructInstance [▶ 1894] type exposes the following members.

Properties

	Name	Description
	_InnerSymbol [▶ 1798]	Gets the inner symbol of this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	AccessRights [▶ 1798]	Gets the access rights. (Inherited from DynamicSymbol [▶ 1791].)
	Attributes [▶ 1799]	Gets the Symbol Attributes (Inherited from DynamicSymbol [▶ 1791].)
	BitSize [▶ 1800]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from DynamicSymbol [▶ 1791].)
	ByteSize [▶ 1800]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from DynamicSymbol [▶ 1791].)
	Category [▶ 1801]	Gets the category. (Inherited from DynamicSymbol [▶ 1791].)
	Comment [▶ 1801]	Gets the comment of the IInstance [▶ 2052] (Inherited from DynamicSymbol [▶ 1791].)
	Connection [▶ 1802]	Gets the connection bound to this DynamicSymbol [▶ 1791] (Inherited from DynamicSymbol [▶ 1791].)
	DataType [▶ 1802]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	HasRpcMethods [▶ 1785]	Gets a value indicating whether this instance has RPC methods (Inherited from DynamicStructInstance [▶ 1779].)
	HasValue [▶ 1803]	Gets a value indicating whether this instance has value (only the non VirtualInstances, what means the Symbols with locations). (Inherited from DynamicSymbol [▶ 1791].)
	InstanceName [▶ 1803]	Gets the name of the instance (without periods (.) (Inherited from DynamicSymbol [▶ 1791].)
	InstancePath [▶ 1804]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from DynamicSymbol [▶ 1791].)
	IsBitType [▶ 1805]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from DynamicSymbol [▶ 1791].)
	IsByteAligned [▶ 1805]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from DynamicSymbol [▶ 1791].)
	IsContainerType [▶ 1806]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from DynamicSymbol [▶ 1791].)
	IsPersistent [▶ 1806]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from DynamicSymbol [▶ 1791].)
	IsPointer [▶ 1807]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsPrimitiveType [▶ 1807]	Gets a value indicating whether this instance is a primitive type. (Inherited from DynamicSymbol [▶ 1791].)
	IsReadOnly [▶ 1808]	Indicates that this instance is read only. (Inherited from DynamicSymbol [▶ 1791].)
	IsRecursive [▶ 1808]	Gets a value indicating whether this instance is recursive. (Inherited from DynamicSymbol [▶ 1791].)
	IsReference [▶ 1809]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from DynamicSymbol [▶ 1791].)
	IsStatic [▶ 1809]	Gets a value indicating whether this instance is static. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	MemberInstances [▶ 1786]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from DynamicStructInstance [▶ 1779].)
	NormalizedName [▶ 1810]	Gets the normalized instance name (fixed name for dynamic property access that doesn't contain invalid characters), (Inherited from DynamicSymbol [▶ 1791].)
	NotificationSettings [▶ 1810]	Gets the notification settings. (Inherited from DynamicSymbol [▶ 1791].)
	Parent [▶ 1811]	Gets the parent Symbol (Inherited from DynamicSymbol [▶ 1791].)
	Size [▶ 1811]	Gets the size of the IInstance [▶ 2052] in bytes. (Inherited from DynamicSymbol [▶ 1791].)
	SubSymbols [▶ 1812]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from DynamicSymbol [▶ 1791].)
	TypeName [▶ 1812]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from DynamicSymbol [▶ 1791].)
	ValueEncoding [▶ 1813]	Gets the value encoding. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicVirtualStructInstance Class [▶ 1894]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.25.2 DynamicVirtualStructInstance Methods

The DynamicVirtualStructInstance [▶ 1894] type exposes the following members.

Methods

	Name	Description
	AddMember [▶ 1900]	Adds an member instance.
	Equals [▶ 1816]	Equals (Inherited from DynamicSymbol [▶ 1791].)
	GetDynamicMemberNames [▶ 1789]	Returns the enumeration of all dynamic member names. (Inherited from DynamicStructInstance [▶ 1779].)
	GetHashCode [▶ 1817]	Gets the HashCode of the Address (Inherited from DynamicSymbol [▶ 1791].)
	ReadAnyValue [▶ 1825]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue. [▶ 1826]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValue(Int32) [▶ 1826]	Reads the Symbols raw value (Inherited from DynamicSymbol [▶ 1791].)
	ReadRawValueAsync [▶ 1827]	Read raw value as an asynchronous operation. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValue. [▶ 1828]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)

	Name	Description
	ReadValue(Int32) [▶ 1831]	Reads the value of this DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	ReadValueAsync [▶ 1832]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously. (Inherited from DynamicSymbol [▶ 1791].)
	ToString [▶ 1832]	Returns a String that represents this instance. (Inherited from DynamicSymbol [▶ 1791].)
	TryGetMember [▶ 1789]	Provides the implementation for operations that get member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as getting a value for a property. (Inherited from DynamicStructInstance [▶ 1779].)
	TryReadValue [▶ 1834]	Reads the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	TrySetMember [▶ 1790]	Provides the implementation for operations that set member values. Classes derived from the DynamicObject class can override this method to specify dynamic behavior for operations such as setting a value for a property. (Inherited from DynamicStructInstance [▶ 1779].)
	TryWriteValue [▶ 1834]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	UpdateAnyValue [▶ 1837]	Reads the value of this Value [▶ 2254] into the specified managed value. (Inherited from DynamicSymbol [▶ 1791].)
	WriteAnyValue [▶ 1838]	Writes the value represented by the managed value to this Value [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte.) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValue(.Byte., Int32) [▶ 1839]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteRawValueAsync [▶ 1840]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object) [▶ 1841]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValue(Object, Int32) [▶ 1844]	Writes the specified value to the DynamicSymbol [▶ 1791]. (Inherited from DynamicSymbol [▶ 1791].)
	WriteValueAsync [▶ 1847]	Writes the Value of the IValueSymbol [▶ 2254] (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicVirtualStructInstance Class [▶ 1894]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.25.2.1 DynamicVirtualStructInstance.AddMember Method

Adds an member instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool AddMember(
    ISymbol memberInstance,
    IVirtualStructInstance parent
)
```

Parameters

memberInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The member instance.
parent	Type: TwinCAT.TypeSystem.IVirtualStructInstance [▶ 2269] The parent struct instance. Usually the this pointer.

Return Value

Type: Boolean

Implements

IVirtualStructInstance.AddMember(ISymbol, IVirtualStructInstance) [▶ 2273]

Reference

DynamicVirtualStructInstance Class [▶ 1894]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.25.3 DynamicVirtualStructInstance Events

The DynamicVirtualStructInstance [▶ 1894] type exposes the following members.

Events

	Name	Description
⚡	RawValueChanged [▶ 1848]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)
⚡	ValueChanged [▶ 1849]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed. (Inherited from DynamicSymbol [▶ 1791].)

Reference

DynamicVirtualStructInstance Class [▶ 1894]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26 EnumValue.T. Class

Enum Value

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.EnumValue.T.
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class EnumValue<T> : IEnumValue
where T : IConvertible
```

Type Parameters

T	Enum base type (byte,sbyte,short,ushort,int,uint,long or ulong)

The EnumValue.T. type exposes the following members.

Properties

	Name	Description
	ManagedBaseType [▶ 1903]	Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)
	Name [▶ 1904]	Gets the name of the Enum Value
	Primitive [▶ 1904]	Gets the value.
	RawValue [▶ 1904]	Gets the raw value of the enumeration (as byte array)
	Size [▶ 1905]	Gets the size of the Enum value (in bytes)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
 	Parse [▶ 1906]	Parse Enum Type string
	ToString [▶ 1907]	Returns a String that represents this instance. (Overrides Object.ToString..)
	TryParse [▶ 1907]	Parse EnumType string

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1 EnumValue.T. Properties

The EnumValue.T. [▶ 1901] generic type exposes the following members.

Properties

	Name	Description
	ManagedBaseType [▶ 1903]	Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)
	Name [▶ 1904]	Gets the name of the Enum Value
	Primitive [▶ 1904]	Gets the value.
	RawValue [▶ 1904]	Gets the raw value of the enumeration (as byte array)
	Size [▶ 1905]	Gets the size of the Enum value (in bytes)

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1.1 EnumValue.T..ManagedBaseType Property

Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Type ManagedBaseType { get; }
```

Property Value

Type: Type
The type of the base.

Implements

IEnumValue.ManagedBaseType [▶ 2029]

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1.2 EnumValue.T..Name Property

Gets the name of the Enum Value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Implements

IEnumValue.Name [▶ 2029]

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1.3 EnumValue.T..Primitive Property

Gets the value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Primitive { get; }
```

Property Value

Type: T [▶ 1901]
The value.

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1.4 EnumValue.T..RawValue Property

Gets the raw value of the enumeration (as byte array)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public byte[] RawValue { get; }
```

Property Value

Type: .Byte.
The raw value.

Implements

IEnumValue.RawValue [▶ 2030]

Exceptions

Exception	Condition
NotSupportedException	Base type of enum is not allowed!

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.1.5 EnumValue.T..Size Property

Gets the size of the Enum value (in bytes)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Size { get; }
```

Property Value

Type: Int32
The size.

Implements

IEnumValue.Size [▶ 2030]

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.2 EnumValue.T. Methods

The EnumValue.T. [▶ 1901] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
S	Parse [▶ 1906]	Parse Enum Type string
	ToString [▶ 1907]	Returns a String that represents this instance. (Overrides Object.ToString..)
S	TryParse [▶ 1907]	Parse EnumType string

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.2.1 EnumValue.T..Parse Method

Parse Enum Type string

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static EnumValue<T> Parse(
    IEnumType<T> type,
    string str
)
```

Parameters

type	Type: TwinCAT.TypeSystem.IEnumType [▶ 2021].T [▶ 1901]. The type.
str	Type: System.String The string.

Return Value

Type: EnumValue [▶ 1901].T [▶ 1901].
EnumValue<T>.

Exceptions

Exception	Condition
FormatException	

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.2.2 EnumValue.T..ToString Method

Returns a String that represents this instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override string ToString()
```

Return Value

Type: String
A String that represents this instance.

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.26.2.3 EnumValue.T..TryParse Method

Parse EnumType string

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool TryParse(
    IEnumType<T> type,
    string str,
    out EnumValue<T> value
)
```

Parameters

type	Type: TwinCAT.TypeSystem.IEnumType [▶ 2021].T [▶ 1901]. The type.
str	Type: System.String The string.
value	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1901].. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

EnumValue.T. Class [▶ 1901]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27 EnumValueCollection Class

Class EnumValueCollection.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.EnumValueCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class EnumValueCollection : IList<IEnumValue>,
    ICollection<IEnumValue>, IEnumerable<IEnumValue>, IEnumerable,
    IEnumValueCollection, IEnumValueCollection<IEnumValue, IConvertible>
```

The EnumValueCollection type exposes the following members.

Properties

	Name	Description
	Count [▶ 1910]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1910]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 1911]	Gets or sets the element at the specified index.
	Item.String. [▶ 1912]	Gets or sets the element at the specified index.

Methods

	Name	Description
	Add [▶ 1914]	Adds an item to the ICollection.T..

	Name	Description
	AsReadOnly [▶ 1915]	Gets as read only.
	Clear [▶ 1915]	Removes all items from the ICollection.T..
	Contains(Object) [▶ 1916]	Determines whether [contains] [the specified value].
	Contains(String) [▶ 1916]	Determines whether [contains] [the specified name].
	Contains(IEnumValue) [▶ 1917]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1918]	Copies the entire list.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1918]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1919]	Gets the names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1919]	Gets the values.
	IndexOf [▶ 1920]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1921]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 1921]	Parses the specified string to the Enum value.
	Remove [▶ 1922]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1923]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInfo [▶ 1923]	Tries the get information.
	TryParse(String, IConvertible.) [▶ 1924]	Parse the specified string to the enum value.
	TryParse(String, IEnumValue.) [▶ 1925]	Parse the specified string to the enum value.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.1 EnumValueCollection Properties

The EnumValueCollection [▶ 1908] type exposes the following members.

Properties

	Name	Description
	Count [▶ 1910]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1910]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 1911]	Gets or sets the element at the specified index.
	Item.String. [▶ 1912]	Gets or sets the element at the specified index.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.1.1 EnumValueCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.1.2 EnumValueCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.1.3 EnumValueCollection.Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 1911]	Gets or sets the element at the specified index.
	Item.String. [▶ 1912]	Gets or sets the element at the specified index.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumValue this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: IEnumValue [▶ 2028]
EnumValue<T>.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

EnumValueCollection Class [▶ 1908]

Item Overload [▶ 1911]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.Item Property (String)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name of the value

Return Value

Type: IConvertible
EnumValue<T>.

Implements

IEnumValueCollection.TEnumValue, TValue..Item.String. [▶ 2035]

Exceptions

Exception	Condition
NotImplementedException	
NotImplementedException	

Reference

EnumValueCollection Class [▶ 1908]

Item Overload [▶ 1911]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2 EnumValueCollection Methods

The EnumValueCollection [▶ 1908] type exposes the following members.

Methods

	Name	Description
	Add [▶ 1914]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 1915]	Gets as read only.
	Clear [▶ 1915]	Removes all items from the ICollection.T..
	Contains(Object) [▶ 1916]	Determines whether [contains] [the specified value].
	Contains(String) [▶ 1916]	Determines whether [contains] [the specified name].
	Contains(IEnumValue) [▶ 1917]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1918]	Copies the entire list.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1918]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1919]	Gets the names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1919]	Gets the values.
	IndexOf [▶ 1920]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1921]	Inserts an item to the IList.T. at the specified index.

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 1921]	Parses the specified string to the Enum value.
	Remove [▶ 1922]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1923]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInfo [▶ 1923]	Tries the get information.
	TryParse(String, IConvertible.) [▶ 1924]	Parse the specified string to the enum value.
	TryParse(String, IEnumValue.) [▶ 1925]	Parse the specified string to the enum value.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.1 EnumValueCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    IEnumValue item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.2 EnumValueCollection.AsReadOnly Method

Gets as read only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyEnumValueCollection AsReadOnly()
```

Field Value

Type: ReadOnlyEnumValueCollection [▶ 2300]
As read only.

Return Value

Type: ReadOnlyEnumValueCollection [▶ 2300]
ReadOnlyEnumValueCollection.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.3 EnumValueCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.4 EnumValueCollection.Contains Method

Overload List

	Name	Description
	Contains(Object) [▶ 1916]	Determines whether [contains] [the specified value].
	Contains(String) [▶ 1916]	Determines whether [contains] [the specified name].
	Contains(IEnumValue) [▶ 1917]	Determines whether the ICollection.T. contains a specific value.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.Contains Method (Object)

Determines whether [contains] [the specified value].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    Object value
)
```

Parameters

value	Type: System.Object The value.

Return Value

Type: Boolean
true if [contains] [the specified value]; otherwise, false.

Reference

EnumValueCollection Class [▶ 1908]

Contains Overload [▶ 1916]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.Contains Method (String)

Determines whether [contains] [the specified name].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

IEnumValueCollection.TEnumValue, TValue..Contains(String) [▶ 2036]

Reference

EnumValueCollection Class [▶ 1908]

Contains Overload [▶ 1916]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.Contains Method (IEnumValue)

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    IEnumValue item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028] The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

EnumValueCollection Class [▶ 1908]

Contains Overload [▶ 1916]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.5 EnumValueCollection.CopyTo Method

Copies the entire list.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    IEnumValue[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.6 EnumValueCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<IEnumValue> GetEnumerator()
```

Return Value

Type: IEnumerator.IEnumValue [▶ 2028].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.7 EnumValueCollection.GetNames Method

Gets the names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetNames. [▶ 2037]

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.8 EnumValueCollection.GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible[] GetValues()
```

Return Value

Type: .IConvertible.
T[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetValues. [▶ 2037]

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.9 EnumValueCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    IEnumValue item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.10 EnumValueCollection.Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    IEnumValue item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.11 EnumValueCollection.Parse Method

Parses the specified string to the Enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: IConvertible
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Parse(String) [▶ 2038]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	name

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.12 EnumValueCollection.Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    IEnumValue item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028] The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.13 EnumValueCollection.RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.14 EnumValueCollection.TryGetInfo Method

Tries the get information.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInfo(
    Object val,
    out IEnumValue ei
)
```

Parameters

val	Type: System.Object The value.

ei	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The ei.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.27.2.15 EnumValueCollection.TryParse Method

Overload List

	Name	Description
	TryParse(String, IConvertible.) [▶ 1924]	Parse the specified string to the enum value.
	TryParse(String, IEnumValue.) [▶ 1925]	Parse the specified string to the enum value.

Reference

EnumValueCollection Class [▶ 1908]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.TryParse Method (String, IConvertible.)

Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string name,
    out IConvertible value
)
```

Parameters

name	Type: System.String The name.
value	Type: System.IConvertible. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

EnumValueCollection Class [▶ 1908]

TryParse Overload [▶ 1924]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.TryParse Method (String, IEnumValue.)

Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string name,
    out IEnumValue value
)
```

Parameters

name	Type: System.String The name.
value	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

EnumValueCollection Class [▶ 1908]

TryParse Overload [▶ 1924]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28 EnumValueCollection.T. Class

Collection of EnumValues [▶ 1901]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.EnumValueCollection.T.
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class EnumValueCollection<T> : IList<EnumValue<T>>,
    ICollection<EnumValue<T>>, IEnumerable<EnumValue<T>>, IEnumerable,
    IEnumValueCollection<EnumValue<T>, T>
where T : IConvertible
```

Type Parameters

T	Base type of enum

The EnumValueCollection.T. type exposes the following members.

Properties

	Name	Description
	Count [▶ 1928]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1928]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 1929]	Gets or sets the element at the specified index.
	Item.String. [▶ 1930]	Gets the enumeration value str from the string representation.

Methods

	Name	Description
	Add [▶ 1932]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 1933]	Gets as read only.
	Clear [▶ 1933]	Removes all items from the ICollection.T..
	Contains(String) [▶ 1934]	Determines whether [contains] [the specified name].
	Contains(T) [▶ 1935]	Determines whether [contains] [the specified value].
	Contains(EnumValue.T.) [▶ 1935]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1936]	Copies the entire list.
S	Empty [▶ 1936]	Return an Empty Collection.

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1937]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1937]	Gets the names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1938]	Gets the values.
	IndexOf [▶ 1938]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1939]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 1940]	Parses the specified string to the Enum value.
	Remove [▶ 1940]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1941]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInfo [▶ 1942]	Tries the get information.
	TryParse(String, EnumValue.T..) [▶ 1943]	Parse the specified string to the enum value.
	TryParse(String, T.) [▶ 1943]	Parse the specified string to the enum value.

Operators

	Name	Description
	.(EnumValueCollection.T. to EnumValueCollection) [▶ 1944]	Performs an explicit conversion from EnumValueCollection.T. to EnumValueCollection [▶ 1908].

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.1 EnumValueCollection.T. Properties

The EnumValueCollection.T. [▶ 1926] generic type exposes the following members.

Properties

	Name	Description
	Count [▶ 1928]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 1928]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 1929]	Gets or sets the element at the specified index.
	Item.String. [▶ 1930]	Gets the enumeration value str from the string representation.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.1.1 EnumValueCollection.T..Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.1.2 EnumValueCollection.T..IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.1.3 EnumValueCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 1929]	Gets or sets the element at the specified index.
	Item.String. [▶ 1930]	Gets the enumeration value str from the string representation.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public EnumValue<T> this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: EnumValue [▶ 1901].T [▶ 1926].
EnumValue<T>.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

EnumValueCollection.T. Class [▶ 1926]

Item Overload [▶ 1929]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..Item Property (String)

Gets the enumeration value str from the string representation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string str
] { get; }
```

Parameters

str	Type: System.String The string.

Return Value

Type: T [▶ 1926]
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Item.String. [▶ 2035]

Exceptions

Exception	Condition
NotImplementedException	

Reference

EnumValueCollection.T. Class [▶ 1926]

Item Overload [▶ 1929]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2 EnumValueCollection.T. Methods

The EnumValueCollection.T. [▶ 1926] generic type exposes the following members.

Methods

	Name	Description
	Add [▶ 1932]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 1933]	Gets as read only.
	Clear [▶ 1933]	Removes all items from the ICollection.T..
	Contains(String) [▶ 1934]	Determines whether [contains] [the specified name].
	Contains(T) [▶ 1935]	Determines whether [contains] [the specified value].
	Contains(EnumValue.T.) [▶ 1935]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 1936]	Copies the entire list.
S	Empty [▶ 1936]	Return an Empty Collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 1937]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 1937]	Gets the names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 1938]	Gets the values.
	IndexOf [▶ 1938]	Determines the index of a specific item in the IList.T..
	Insert [▶ 1939]	Inserts an item to the IList.T. at the specified index.

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 1940]	Parses the specified string to the Enum value.
	Remove [▶ 1940]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 1941]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInfo [▶ 1942]	Tries the get information.
	TryParse(String, EnumValue.T..) [▶ 1943]	Parse the specified string to the enum value.
	TryParse(String, T.) [▶ 1943]	Parse the specified string to the enum value.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.1 EnumValueCollection.T..Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    EnumValue<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926]. The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.2 EnumValueCollection.T..AsReadOnly Method

Gets as read only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyEnumValueCollection<T> AsReadOnly()
```

Field Value

Type: ReadOnlyEnumValueCollection [▶ 2309].T [▶ 1926].
As read only.

Return Value

Type: ReadOnlyEnumValueCollection [▶ 2309].T [▶ 1926].
ReadOnlyEnumValueCollection<T>.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.3 EnumValueCollection.T..Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.4 EnumValueCollection.T..Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 1934]	Determines whether [contains] [the specified name].
	Contains(T) [▶ 1935]	Determines whether [contains] [the specified value].
	Contains(EnumValue.T.) [▶ 1935]	Determines whether the ICollection.T. contains a specific value.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..Contains Method (String)

Determines whether [contains] [the specified name].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

IEnumValueCollection.TEnumValue, TValue..Contains(String) [▶ 2036]

Reference

EnumValueCollection.T. Class [▶ 1926]

Contains Overload [▶ 1934]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..Contains Method (T)

Determines whether [contains] [the specified value].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    T value
)
```

Parameters

value	Type: T [▶ 1926] The value.

Return Value

Type: Boolean
true if [contains] [the specified value]; otherwise, false.

Reference

EnumValueCollection.T. Class [▶ 1926]

Contains Overload [▶ 1934]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..Contains Method (EnumValue.T.)

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    EnumValue<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926]. The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

EnumValueCollection.T. Class [▶ 1926]

Contains Overload [▶ 1934]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.5 EnumValueCollection.T..CopyTo Method

Copies the entire list.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    EnumValue<T>[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926].. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.6 EnumValueCollection.T..Empty Method

Return an Empty Collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static EnumValueCollection<T> Empty()
```

Return Value

Type: EnumValueCollection [▶ 1926].T [▶ 1926].
EnumValueCollection<T>.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.7 EnumValueCollection.T..GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<EnumValue<T>> GetEnumerator()
```

Return Value

Type: IEnumerator.EnumValue [▶ 1901].T [▶ 1926]..
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.8 EnumValueCollection.T..GetNames Method

Gets the names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetNames. [▶ 2037]

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.9 EnumValueCollection.T..GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T[] GetValues()
```

Return Value

Type: .T [▶ 1926].
T[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetValues. [▶ 2037]

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.10 EnumValueCollection.T..IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    EnumValue<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926]. The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.11 EnumValueCollection.T..Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    EnumValue<T> item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926]. The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.12 EnumValueCollection.T..Parse Method

Parses the specified string to the Enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 1926]
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Parse(String) [▶ 2038]

Exceptions

Exception	Condition
ArgumentOutOfRangeException	name

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.13 EnumValueCollection.T..Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    EnumValue<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926]. The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Exceptions

Exception	Condition
NotImplementedException	

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.14 EnumValueCollection.T..RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.15 EnumValueCollection.T..TryGetInfo Method

Tries the get information.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInfo(
    T val,
    out EnumValue<T> ei
)
```

Parameters

val	Type: T [▶ 1926] The value.
ei	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926].. The ei.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.2.16 EnumValueCollection.T..TryParse Method

Overload List

	Name	Description
	TryParse(String, EnumValue.T..) [▶ 1943]	Parse the specified string to the enum value.
	TryParse(String, T.) [▶ 1943]	Parse the specified string to the enum value.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..TryParse Method (String, EnumValue.T..)

Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string name,
    out EnumValue<T> value
)
```

Parameters

name	Type: System.String The name.
value	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 1926].. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

EnumValueCollection.T. Class [▶ 1926]

TryParse Overload [▶ 1942]

TwinCAT.TypeSystem Namespace [▶ 1622]

EnumValueCollection.T..TryParse Method (String, T.)

Parse the specified string to the enum value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string name,
    out T value
)
```

Parameters

name	Type: System.String The name.
value	Type: T [▶ 1926]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

EnumValueCollection.T. Class [▶ 1926]

TryParse Overload [▶ 1942]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.3 EnumValueCollection.T. Type Conversions

The EnumValueCollection.T. [▶ 1926] generic type exposes the following members.

Operators

	Name	Description
 	. (EnumValueCollection.T. to EnumValueCollection) [▶ 1944]	Performs an explicit conversion from EnumValueCollection.T. [▶ 1926] to EnumValueCollection [▶ 1908].

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.28.3.1 EnumValueCollection.T. . Conversion (EnumValueCollection.T. to EnumValueCollection)

Performs an explicit conversion from EnumValueCollection.T. [▶ 1926] to EnumValueCollection [▶ 1908].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static explicit operator EnumValueCollection (
    EnumValueCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.EnumValueCollection [▶ 1926].T [▶ 1926]. The coll.

Return Value

Type: EnumValueCollection [▶ 1908]
The result of the conversion.

Reference

EnumValueCollection.T. Class [▶ 1926]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29 FieldCollection Class

Collection of IField [▶ 2040] objects.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection [▶ 2460].IField [▶ 2040].
    TwinCAT.TypeSystem.FieldCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class FieldCollection : InstanceCollection<IField>,
    IFieldCollection, IInstanceCollection<IField>, IList<IField>,
    ICollection<IField>, IEnumerable<IField>, IEnumerable
```

The FieldCollection type exposes the following members.

Constructors

	Name	Description
	FieldCollection. [▶ 1947]	Initializes a new instance of the FieldCollection class.

	Name	Description
	FieldCollection(IEnumerable.IField.) [▶ 1948]	Initializes a new instance of the FieldCollection class (copy constructor)

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the LIst of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 1950]	Returns a read only copy of this collection (shallow copy)
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 1951]	Clones this FieldCollection.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetMember [▶ 1951]	Tries to get the specified member

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.1 FieldCollection Constructor

Overload List

	Name	Description
	FieldCollection. [▶ 1947]	Initializes a new instance of the FieldCollection [▶ 1945] class.
	FieldCollection(IEnumerable.IField.) [▶ 1948]	Initializes a new instance of the FieldCollection [▶ 1945] class (copy constructor)

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.1.1 FieldCollection Constructor

Initializes a new instance of the FieldCollection [▶ 1945] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public FieldCollection()
```

Reference

FieldCollection Class [▶ 1945]

FieldCollection Overload [▶ 1947]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.1.2 FieldCollection Constructor (IEnumerable.IField.)

Initializes a new instance of the FieldCollection [▶ 1945] class (copy constructor)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public FieldCollection(
    IEnumerable<IField> coll
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.IField [▶ 2040]. The coll.

Reference

FieldCollection Class [▶ 1945]

FieldCollection Overload [▶ 1947]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.2 FieldCollection Properties

The FieldCollection [▶ 1945] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the IList of instances. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.3 FieldCollection Methods

The FieldCollection [▶ 1945] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 1950]	Returns a read only copy of this collection (shallow copy)
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 1951]	Clones this FieldCollection [▶ 1945].
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetMember [▶ 1951]	Tries to get the specified member

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.3.1 FieldCollection.AsReadOnly Method

Returns a read only copy of this collection (shallow copy)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyFieldCollection AsReadOnly()
```

Return Value

Type: ReadOnlyFieldCollection [▶ 2317]
The readonly copy.

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.3.2 FieldCollection.Clone Method

Clones this FieldCollection [▶ 1945].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public FieldCollection Clone()
```

Return Value

Type: FieldCollection [▶ 1945]
A cloned FieldCollection [▶ 1945].

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.29.3.3 FieldCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMember(
    string fieldName,
    out IField symbol
)
```

Parameters

fieldName	Type: System.String Name of the member.
symbol	Type: TwinCAT.TypeSystem.IField [▶ 2040]. The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IFieldCollection.TryGetMember(String, IField.) [▶ 2045]

Reference

FieldCollection Class [▶ 1945]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.30 IAliasInstance Interface

Interface representing an instance of an IAliasType [▶ 1954].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAliasInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IAliasInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

6.11.30.1 IAliasInstance Properties

The IAliasInstance [▶ 1952] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IAliasInstance Interface [▶ 1952]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.31 IAliasType Interface

Interface representing an Alias Type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAliasType : IDataType,
    IBitSize
```

The IAliasType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type
	BaseTypeName [▶ 1957]	Gets the BaseType name
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.31.1 IAliasType Properties

The IAliasType [▶ 1954] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type
	BaseTypeName [▶ 1957]	Gets the BaseType name
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IAliasType Interface [▶ 1954]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.31.1.1 IAliasType.BaseType Property

Gets the Base Type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The alias base type or **null** if not resolved.

Reference

IAliasType Interface [▶ 1954]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.31.1.2 IAliasType.BaseTypeName Property

Gets the BaseType name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string BaseTypeName { get; }
```

Property Value

Type: String

Reference

IAliasType Interface [▶ 1954]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.32 IAnyTypeMarshaler Interface

Interface IAnyTypeMarshaler Implements the IGenericTypeMarshaler [▶ 2046]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAnyTypeMarshaler : IGenericTypeMarshaler,
    ITypeMarshaler
```

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Object, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.
	CanMarshal(Type, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.
	Marshal(Object, Encoding, Span) [▶ 2219]	(Inherited from ITypeMarshaler [▶ 2216].)
	Marshal(Object, .Int32., Encoding, Span) [▶ 1961]	
	MarshalSize(Object, Encoding) [▶ 2219]	Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [▶ 2216].)
	MarshalSize(Object, .Int32., Encoding) [▶ 1962]	Gets the byte size of the value when marshalled.
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal(Type, .Int32., ReadOnlySpan, Void, Byte) [▶ 1964]	
	Unmarshal.T.(ReadOnlySpan, Void, Byte) [▶ 2048]	(Inherited from IGenericTypeMarshaler [▶ 2046].)

Remarks

The IAnyTypeMarshaler supports value marshalling / Unmarshalling with the ADS ANY_TYPE concept, what means that the specified managed type is supported by an arguments metadata array (args parameter).

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IGenericTypeMarshaler [▶ 2046]

6.11.32.1 IAnyTypeMarshaler Methods

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Object, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.
	CanMarshal(Type, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.
	Marshal(Object, Encoding, Span) [▶ 2219]	(Inherited from ITypeMarshaler [▶ 2216].)
	Marshal(Object, .Int32., Encoding, Span) [▶ 1961]	
	MarshalSize(Object, Encoding) [▶ 2219]	Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [▶ 2216].)
	MarshalSize(Object, .Int32., Encoding) [▶ 1962]	Gets the byte size of the value when marshalled.
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal(Type, .Int32., ReadOnlySpan, Void, Byte) [▶ 1964]	
	Unmarshal.T.(ReadOnlySpan, Void, Byte) [▶ 2048]	(Inherited from IGenericTypeMarshaler [▶ 2046].)

Reference

IAnyTypeMarshaler Interface [▶ 1957]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.32.1.1 IAnyTypeMarshaler.CanMarshal Method

Overload List

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [▶ 2216].)

	Name	Description
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Object, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.
	CanMarshal(Type, .Int32.) [▶ 1960]	Determines whether ADS can marshal the specified managed data type.

Reference

IAnyTypeMarshaler Interface [▶ 1957]

TwinCAT.TypeSystem Namespace [▶ 1622]

IAnyTypeMarshaler.CanMarshal Method (Object, .Int32.)

Determines whether ADS can marshal the specified managed data type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool CanMarshal(
    Object value,
    int[] args
)
```

Parameters

value	Type: System.Object The managed value.
args	Type: .System.Int32. The arguments.

Return Value

Type: Boolean
true if this instance can marshal the specified managed type; otherwise, false.

Reference

IAnyTypeMarshaler Interface [▶ 1957]

CanMarshal Overload [▶ 1959]

TwinCAT.TypeSystem Namespace [▶ 1622]

IAnyTypeMarshaler.CanMarshal Method (Type, .Int32.)

Determines whether ADS can marshal the specified managed data type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool CanMarshal(
    Type anyType,
    int[] args
)
```

Parameters

anyType	Type: System.Type The Managed data type.
args	Type: .System.Int32. The arguments.

Return Value

Type: Boolean
true if this instance can marshal the specified managed type; otherwise, false.

Reference

IAnyTypeMarshaler Interface [▶ 1957]

CanMarshal Overload [▶ 1959]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.32.1.2 IAnyTypeMarshaler.Marshal Method

Overload List

	Name	Description
	Marshal(Object, Encoding, Span) [▶ 2219]	(Inherited from ITypeMarshaler [▶ 2216].)
	Marshal(Object, .Int32., Encoding, Span) [▶ 1961]	

Reference

IAnyTypeMarshaler Interface [▶ 1957]

TwinCAT.TypeSystem Namespace [▶ 1622]

IAnyTypeMarshaler.Marshal Method (Object, .Int32., Encoding, Span`1)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Marshal(
    Object anyValue,
    int[] args,
    Encoding encoding,
    Span destination
)
```

Parameters

anyValue	Type: System.Object
args	Type: .System.Int32.
encoding	Type: System.Text.Encoding
destination	Type: Span

Return Value

Type: Int32

Reference

IAnyTypeMarshaler Interface [▶ 1957]

Marshal Overload [▶ 1961]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.32.1.3 IAnyTypeMarshaler.MarshalSize Method

Overload List

	Name	Description
	MarshalSize(Object, Encoding) [▶ 2219]	Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [▶ 2216].)
	MarshalSize(Object, .Int32., Encoding) [▶ 1962]	Gets the byte size of the value when marshalled.

Reference

IAnyTypeMarshaler Interface [▶ 1957]

TwinCAT.TypeSystem Namespace [▶ 1622]

IAnyTypeMarshaler.MarshalSize Method (Object, .Int32., Encoding)

Gets the byte size of the value when marshalled.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int MarshalSize(
    Object anyValue,
    int[] args,
    Encoding encoding
)
```

Parameters

anyValue	Type: System.Object Any value.
args	Type: .System.Int32. The arguments.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: Int32
The marshal size of the value.

Reference

IAnyTypeMarshaler Interface [▶ 1957]

MarshalSize Overload [▶ 1962]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.32.1.4 IAnyTypeMarshaler.Unmarshal Method

Overload List

	Name	Description
	Unmarshal.T.(ReadOnlySpan, Void, Byte) [▶ 2048]	(Inherited from IGenericTypeMarshaler [▶ 2046].)
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal(Type, .Int32., ReadOnlySpan, Void, Byte) [▶ 1964]	

Reference

IAnyTypeMarshaler Interface [▶ 1957]

TwinCAT.TypeSystem Namespace [▶ 1622]

IAnyTypeMarshaler.Unmarshal Method (Type, .Int32., ReadOnlySpan`1, Void, Byte)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Unmarshal(
    Type anyType,
    int[] args,
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters

anyType	Type: System.Type
args	Type: .System.Int32.
source	Type: ReadOnlySpan
encoding	Type: System.Void
value	Type: System.Byte

Return Value

Type: Int32

Reference

IAnyTypeMarshaler Interface [▶ 1957]

Unmarshal Overload [▶ 1963]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33 IArrayInstance Interface

Interface representing an array instance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IArrayInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IArrayInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	Dimensions [▶ 1967]	Gets the dimensions as read only collection.
	Elements [▶ 1968]	Gets the contained Array Elements as read only collection.
	ElementType [▶ 1968]	Gets the type of the contained elements.
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Item [▶ 1969]	Gets the ISymbol [▶ 2176] with the specified indices.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

	Name	Description
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	TryGetElement(IList.. Int32.., ISymbol.) [▶ 1970]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1971]	Tries to get the array element with specified indices (only first level on jagged arrays)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.1 IArrayInstance Properties

The IArrayInstance [▶ 1964] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	Dimensions [▶ 1967]	Gets the dimensions as read only collection.
	Elements [▶ 1968]	Gets the contained Array Elements as read only collection.
	ElementType [▶ 1968]	Gets the type of the contained elements.
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)

	Name	Description
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Item [▶ 1969]	Gets the ISymbol [▶ 2176] with the specified indices.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.1.1 IArrayInstance.Dimensions Property

Gets the dimensions as read only collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDimensionCollection Dimensions { get; }
```

Property Value

Type: IDimensionCollection [▶ 2000]
The dimensions.

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.1.2 IArrayInstance.Elements Property

Gets the contained Array Elements as read only collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> Elements { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The elements.

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.1.3 IArrayInstance.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ElementType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the element.

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.1.4 IArrayInstance.Item Property

Gets the ISymbol [▶ 2176] with the specified indices.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol this[
    int[] indices
] { get; }
```

Parameters

indices	Type: .System.Int32. The indices.

Property Value

Type: ISymbol [▶ 2176]
The ISymbol [▶ 2176].

Return Value

Type: ISymbol [▶ 2176]

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.2 IArrayInstance Methods

Methods

	Name	Description
	TryGetElement(IList.. Int32.., ISymbol.) [▶ 1970]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1971]	Tries to get the array element with specified indices (only first level on jagged arrays)

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.33.2.1 IArrayInstance.TryGetElement Method

Overload List

	Name	Description
	TryGetElement(IList..Int32.., ISymbol.) [▶ 1970]	Tries to get the array element with the specified indices (jagged array support).
	TryGetElement(.Int32., ISymbol.) [▶ 1971]	Tries to get the array element with specified indices (only first level on jagged arrays)

Reference

IArrayInstance Interface [▶ 1964]

TwinCAT.TypeSystem Namespace [▶ 1622]

IArrayInstance.TryGetElement Method (IList..Int32.., ISymbol.)

Tries to get the array element with the specified indices (jagged array support).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetElement(
    IList<int[]> jaggedIndices,
    out ISymbol symbol
)
```

Parameters

jaggedIndices	Type: System.Collections.Generic.IList..Int32.. The jagged indices list.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]. The symbol.

Return Value

Type: Boolean
true if found, false if the jagged indices specifiers is out-of-range.

Reference

IArrayInstance Interface [▶ 1964]

TryGetElement Overload [▶ 1970]

TwinCAT.TypeSystem Namespace [▶ 1622]

IArrayInstance.TryGetElement Method (.Int32., ISymbol.)

Tries to get the array element with specified indices (only first level on jagged arrays)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetElement(
    int[] indices,
    out ISymbol symbol
)
```

Parameters

indices	Type: .System.Int32. The indices.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]. The found Array element symbol (out-parameter).

Return Value

Type: Boolean
true if found, false if the indices specifiers is out-of-range.

Reference

IArrayInstance Interface [▶ 1964]

TryGetElement Overload [▶ 1970]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34 IArrayType Interface

Interface representing an array DataType [▶ 1986].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IArrayType : IDataType,
    IBitSize
```

The IArrayType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Dimensions [▶ 1974]	Gets the dimensions as read only collection.
	ElementType [▶ 1974]	Gets the type of the contained elements.
	ElementTypeName [▶ 1974]	Gets the name of the element datatype.
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsJagged [▶ 1975]	Gets a value indicating whether this instance is jagged.
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	JaggedLevel [▶ 1975]	Gets the jagged level (Non-Jagged Arrays have level 1)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1 IArrayType Properties

The IArrayType [▶ 1971] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Dimensions [▶ 1974]	Gets the dimensions as read only collection.
	ElementType [▶ 1974]	Gets the type of the contained elements.
	ElementTypeName [▶ 1974]	Gets the name of the element datatype.
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsJagged [▶ 1975]	Gets a value indicating whether this instance is jagged.
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	JaggedLevel [▶ 1975]	Gets the jagged level (Non-Jagged Arrays have level 1)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1.1 IArrayType.Dimensions Property

Gets the dimensions as read only collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDimensionCollection Dimensions { get; }
```

Property Value

Type: IDimensionCollection [▶ 2000]
The dimensions.

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1.2 IArrayType.ElementType Property

Gets the type of the contained elements.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ElementType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the element or **null** if not resolved.

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1.3 IArrayType.ElementTypeName Property

Gets the name of the element datatype.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ElementTypeName { get; }
```

Property Value

Type: String
The name of the element datatype.

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1.4 IArrayType.IsJagged Property

Gets a value indicating whether this instance is jagged.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsJagged { get; }
```

Property Value

Type: Boolean
true if this instance is jagged; otherwise, false.

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.34.1.5 IArrayType.JaggedLevel Property

Gets the jagged level (Non-Jagged Arrays have level 1)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int JaggedLevel { get; }
```

Property Value

Type: Int32
The jagged level.

Reference

IArrayType Interface [▶ 1971]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.35 IArrayValue Interface

Interface IArrayValue

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IArrayValue : IValue
```

The IArrayValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TryGetArrayElementValues [▶ 1978]	Returns Array Element values.
	TryGetIndexValue [▶ 1978]	Tries to get the specified Array Element
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)

	Name	Description
	TrySetIndexValue [▶ 1979]	Tries to set the indexed value on Arrays
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IValue [▶ 2226]

6.11.35.1 IArrayValue Properties

The IArrayValue [▶ 1976] type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Reference

IArrayValue Interface [▶ 1976]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.35.2 IArrayValue Methods

The IArrayValue [▶ 1976] type exposes the following members.

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)

	Name	Description
	TryGetArrayElementValues [▶ 1978]	Returns Array Element values.
	TryGetIndexValue [▶ 1978]	Tries to get the specified Array Element
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TrySetIndexValue [▶ 1979]	Tries to set the indexed value on Arrays
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

IArrayValue Interface [▶ 1976]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.35.2.1 IArrayValue.TryGetArrayElementValues Method

Returns Array Element values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetArrayElementValues(
    out IEnumerable<Object> elementValues
)
```

Parameters

elementValues	Type: System.Collections.Generic.IEnumerable.Object.. The element values.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

IArrayValue Interface [▶ 1976]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.35.2.2 IArrayValue.TryGetIndexValue Method

Tries to get the specified Array Element

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetIndexValue(
    int[] indices,
    out Object value
)
```

Parameters

indices	Type: .System.Int32. The indices.
value	Type: System.Object. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

IArrayValue Interface [▶ 1976]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.35.2.3 IArrayValue.TrySetIndexValue Method

Tries to set the indexed value on Arrays

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TrySetIndexValue(
    Object[] indexes,
    Object value
)
```

Parameters

indexes	Type: .System.Object. The indexes.
value	Type: System.Object The value.

Return Value

Type: Boolean
true if succeeded, false otherwise.

Reference

IArrayValue Interface [▶ 1976]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.36 IAttributedInstance Interface

Interface IAttributedInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAttributedInstance : IInstance,
    IBitSize
```

The IAttributedInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

	Name	Description
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IInstance [▶ 2052]

6.11.36.1 IAttributedInstance Properties

The IAttributedInstance [▶ 1980] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding.

Reference

IAttributedInstance Interface [▶ 1980]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.36.1.1 IAttributedInstance.Attributes Property

Gets the Type Attributes.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection [▶ 2211]
The attributes.

Reference

IAttributedInstance Interface [▶ 1980]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.36.1.2 IAttributedInstance.ValueEncoding Property

Gets the value encoding.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Encoding ValueEncoding { get; }
```

Property Value

Type: Encoding
The value encoding.

Reference

IAttributedInstance Interface [▶ 1980]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37 IBitSize Interface

Interface IBitSize

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IBitSize
```

The IBitSize type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits.
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984]

Remarks

Specifies the Bitness of the the object and the bit resp. byte sizes.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1 IBitSize Properties

The IBitSize [▶ 1982] type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits.
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984]

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1.1 IBitSize.BitSize Property

Gets the size of the IDataType [▶ 1986] in bits.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int BitSize { get; }
```

Property Value

Type: Int32
The size of IDataType [▶ 1986] in bits.

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1.2 IBitSize.ByteSize Property

Gets the (aligned) size of of the Type/Instance in Bytes

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ByteSize { get; }
```

Property Value

Type: Int32
The size of the byte.

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1.3 IBitSize.IsBitType Property

Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsBitType { get; }
```

Property Value

Type: Boolean
true if this instance is bit mapping; otherwise, false.

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1.4 IBitSize.IsByteAligned Property

Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsByteAligned { get; }
```

Property Value

Type: Boolean
true if this instance is byte aligned; otherwise, false.

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.37.1.5 IBitSize.Size Property

Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Size { get; }
```

Property Value

Type: Int32
The size of the bit.

Reference

IBitSize Interface [▶ 1982]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38 IDataType Interface

Base interface for objects representing data types

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDataType : IBitSize
```

The IDataType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType
	BitSize [▶ 1984]	Gets the size of the IDataType in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category
	Comment [▶ 1988]	Gets the comment behind the variable declaration.
	FullName [▶ 1989]	Gets the full name of the IDataType (Namespace + Name)
	Id [▶ 1989]	Gets the ID of the DataType
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType is a container type
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType is a pointer type
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType is primitive
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType is a reference type
	Name [▶ 1992]	Gets the name of the Data Type (without namespace)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType exists.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1 IDataType Properties

The IDataType [▶ 1986] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986]
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category
	Comment [▶ 1988]	Gets the comment behind the variable declaration.
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name)
	Id [▶ 1989]	Gets the ID of the DataType
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type
	Name [▶ 1992]	Gets the name of the Data Type (without namespace)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.1 IDataType.Attributes Property

Gets the attributes of the IDataType [▶ 1986]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ITypeAttributeCollection Attributes { get; }
```

Property Value

Type: ITypeAttributeCollection [▶ 2211]
The attributes.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.2 IDataType.Category Property

Gets the Data Type category

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DataTypeCategory Category { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The category.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.3 IDataType.Comment Property

Gets the comment behind the variable declaration.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Comment { get; }
```

Property Value

Type: String
Comment behind the variable declaration.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.4 IDataType.FullName Property

Gets the full name of the IDataType [▶ 1986] (Namespace + Name)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string FullName { get; }
```

Property Value

Type: String
The full name.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.5 IDataType.Id Property

Gets the ID of the DataType

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Id { get; }
```

Property Value

Type: Int32
The id.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.6 IDataType.IsContainer Property

Gets a value indicating whether this IDataType [▶ 1986] is a container type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsContainer { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Remarks

Container Types are all types that contain SubElements like

- Array [▶ 1649]
- Pointer [▶ 1649]
- Union [▶ 1649]
- Struct [▶ 1649]
- Function [▶ 1649]
- FunctionBlock [▶ 1649]
- Program [▶ 1649]

and the Alias [▶ 1649] and Reference [▶ 1649] types, if they have a container type as base type.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

IDataType.Category [▶ 1988]

6.11.38.1.7 IDataType.IsPointer Property

Gets a value indicating whether this IDataType [▶ 1986] is a pointer type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPointer { get; }
```

Property Value

Type: Boolean
true if this instance is pointer type; otherwise, false.

Remarks

Pointer types can be dereferenced with the '^' operator.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

IDataType.Category [▶ 1988]

6.11.38.1.8 IDataType.IsPrimitive Property

Gets a value indicating whether this IDataType [▶ 1986] is primitive

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.9 IDataType.IsReference Property

Gets a value indicating whether this IDataType [▶ 1986] is a reference type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsReference { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Remarks

Reference types can be dereferenced.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

IDataType.Category [▶ 1988]

6.11.38.1.10 IDataType.Name Property

Gets the name of the Data Type (without namespace)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.38.1.11 IDataType.Namespace Property

Gets the namespace string within the IDataType [▶ 1986] exists.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Namespace { get; }
```

Property Value

Type: String
The namespace.

Reference

IDataType Interface [▶ 1986]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.39 IDataTypeCollection Interface

Interface IDataTypeCollection Implements the ICollection.T.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDataTypeCollection : IDataTypeCollection<IDataType>,
    ICollection<IDataType>, IEnumerable<IDataType>, IEnumerable
```

The IDataTypeCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IDataType [▶ 1986]..)
	Item [▶ 1996]	Gets the IDataType [▶ 1986] with the specified name. (Inherited from IDataTypeCollection.T. [▶ 1995].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IDataType [▶ 1986]..)
	ContainsType [▶ 1997]	Determines whether the container contains the specified IDataType [▶ 1986]. (Inherited from IDataTypeCollection.T. [▶ 1995].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IDataType [▶ 1986]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IDataType [▶ 1986]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	TryGetType [▶ 1998]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995]. (Inherited from IDataTypeCollection.T. [▶ 1995].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.Collections.Generic.ICollection.T.

6.11.39.1 IDataTypeCollection Properties

The IDataTypeCollection [▶ 1993] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IDataType [▶ 1986]..)
	Item [▶ 1996]	Gets the IDataType [▶ 1986] with the specified name. (Inherited from IDataTypeCollection.T. [▶ 1995].)

Reference

IDataTypeCollection Interface [▶ 1993]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.39.2 IDataTypeCollection Methods

The IDataTypeCollection [▶ 1993] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IDataType [▶ 1986]..)
	ContainsType [▶ 1997]	Determines whether the container contains the specified IDataType [▶ 1986]. (Inherited from IDataTypeCollection.T. [▶ 1995].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IDataType [▶ 1986]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IDataType [▶ 1986]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IDataType [▶ 1986]..)
	TryGetType [▶ 1998]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995]. (Inherited from IDataTypeCollection.T. [▶ 1995].)

Reference

IDataTypeCollection Interface [▶ 1993]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40 IDataTypeCollection.T. Interface

Data Type container interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDataTypeCollection<T> : ICollection<T>,
    IEnumerable<T>, IEnumerable
where T : class, IDataType
```

Type Parameters

T	Data Type type.

The IDataTypeCollection.T. type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T..)
	Item [▶ 1996]	Gets the IDataType [▶ 1986] with the specified name.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T..)
	ContainsType [▶ 1997]	Determines whether the container contains the specified IDataType [▶ 1986].
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T..)
	TryGetType [▶ 1998]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T..

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40.1 IDataTypeCollection.T. Properties

The IDataTypeCollection.T. [▶ 1995] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T [▶ 1995]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T [▶ 1995]..)
	Item [▶ 1996]	Gets the IDataType [▶ 1986] with the specified name.

Reference

IDataTypeCollection.T. Interface [▶ 1995]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40.1.1 IDataTypeCollection.T..Item Property

Gets the IDataType [▶ 1986] with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 1995]
T.

Reference

IDataTypeCollection.T. Interface [▶ 1995]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40.2 IDataTypeCollection.T. Methods

The IDataTypeCollection.T. [▶ 1995] generic type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T [▶ 1995]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T [▶ 1995]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T [▶ 1995]..)
	ContainsType [▶ 1997]	Determines whether the container contains the specified IDataType [▶ 1986].
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T [▶ 1995]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T [▶ 1995]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T [▶ 1995]..)
	TryGetType [▶ 1998]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995].

Reference

IDataTypeCollection.T. Interface [▶ 1995]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40.2.1 IDataTypeCollection.T..ContainsType Method

Determines whether the container contains the specified IDataType [▶ 1986].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool ContainsType(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if contained; otherwise, false.

Reference

IDataTypeCollection.T. Interface [▶ 1995]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.40.2.2 IDataTypeCollection.T..TryGetType Method

Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetType(
    string name,
    out T type
)
```

Parameters

name	Type: System.String The name.
type	Type: T [▶ 1995]. The type (Out parameter)

Return Value

Type: Boolean
true if found

Reference

IDataTypeCollection.T. Interface [▶ 1995]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.41 IDimension Interface

Interface representing a single Dimension of an ArrayType [▶ 1971].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDimension
```

The IDimension type exposes the following members.

Properties

	Name	Description
	ElementCount [▶ 1999]	Gets the number of elements within that IDimension.
	LowerBound [▶ 1999]	Gets the lower bound of elements within that IDimension.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.41.1 IDimension Properties

The IDimension [▶ 1998] type exposes the following members.

Properties

	Name	Description
	ElementCount [▶ 1999]	Gets the number of elements within that IDimension [▶ 1998].
	LowerBound [▶ 1999]	Gets the lower bound of elements within that IDimension [▶ 1998].

Reference

IDimension Interface [▶ 1998]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.41.1.1 IDimension.ElementCount Property

Gets the number of elements within that IDimension [▶ 1998].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ElementCount { get; }
```

Property Value

Type: Int32
The element count.

Reference

IDimension Interface [▶ 1998]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.41.1.2 IDimension.LowerBound Property

Gets the lower bound of elements within that IDimension [▶ 1998].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int LowerBound { get; }
```

Property Value

Type: Int32
The lower bound.

Reference

IDimension Interface [▶ 1998]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42 IDimensionCollection Interface

Interface IDimensionCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDimensionCollection : IList<IDimension>,
    ICollection<IDimension>, IEnumerable<IDimension>, IEnumerable
```

The IDimensionCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	ElementCount [▶ 2002]	Gets the Number of elements in all Dimensions
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IDimension [▶ 1998]..)
	Item	Gets or sets the element at the specified index. (Inherited from IList.IDimension [▶ 1998]..)
	LowerBounds [▶ 2002]	Gets the lower bounds.
	UpperBounds [▶ 2002]	Gets the lower bounds.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)

	Name	Description
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IDimension [▶ 1998]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IDimension [▶ 1998]..)
	GetDimensionLengths [▶ 2004]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IDimension [▶ 1998]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IDimension [▶ 1998]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IDimension [▶ 1998]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IDimension [▶ 1998]..)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IDimension [▶ 1998]

6.11.42.1 IDimensionCollection Properties

The IDimensionCollection [▶ 2000] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	ElementCount [▶ 2002]	Gets the Number of elements in all Dimensions
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IDimension [▶ 1998]..)
	Item	Gets or sets the element at the specified index. (Inherited from IList.IDimension [▶ 1998]..)
	LowerBounds [▶ 2002]	Gets the lower bounds.
	UpperBounds [▶ 2002]	Gets the lower bounds.

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42.1.1 IDimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ElementCount { get; }
```

Property Value

Type: Int32

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42.1.2 IDimensionCollection.LowerBounds Property

Gets the lower bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int[] LowerBounds { get; }
```

Property Value

Type: .Int32.
The lower bounds.

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42.1.3 IDimensionCollection.UpperBounds Property

Gets the lower bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int[] UpperBounds { get; }
```

Property Value

Type: .Int32.
The lower bounds.

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42.2 IDimensionCollection Methods

The IDimensionCollection [▶ 2000] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IDimension [▶ 1998]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IDimension [▶ 1998]..)
	GetDimensionLengths [▶ 2004]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IDimension [▶ 1998]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IDimension [▶ 1998]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IDimension [▶ 1998]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IDimension [▶ 1998]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IDimension [▶ 1998]..)

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.42.2.1 IDimensionCollection.GetDimensionLengths Method

Gets an array the specifies the Lengths of each Array Dimension

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int[] GetDimensionLengths()
```

Return Value

Type: .Int32.
System.Int32[].

Reference

IDimensionCollection Interface [▶ 2000]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.43 IDynamicSymbol Interface

Interface IDynamicSymbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDynamicSymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IDynamicSymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)

	Name	Description
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	NormalizedName [▶ 2007]	Gets the normalized instance name (fixed name for dynamic property access)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.43.1 IDynamicSymbol Properties

The IDynamicSymbol [▶ 2004] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	NormalizedName [▶ 2007]	Gets the normalized instance name (fixed name for dynamic property access)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)

	Name	Description
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IDynamicSymbol Interface [▶ 2004]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.43.1.1 IDynamicSymbol.NormalizedName Property

Gets the normalized instance name (fixed name for dynamic property access)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string NormalizedName { get; }
```

Property Value

Type: String
The normalized instance name (can be the same like InstanceName [▶ 2054]

Reference

IDynamicSymbol Interface [▶ 2004]

TwinCAT.TypeSystem Namespace [▶ 1622]

IInstance.InstanceName [▶ 2054]

ISymbolFactory.InvalidCharacters [▶ 2190]

6.11.44 IDynamicSymbolLoader Interface

Dynamic symbol loader interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDynamicSymbolLoader : ISymbolLoader,
    ISymbolProvider, ISymbolServer
```

The IDynamicSymbolLoader type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types. (Inherited from ISymbolLoader [▶ 2200].)
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)
	Settings [▶ 2202]	Gets or sets the access Method (Inherited from ISymbolLoader [▶ 2200].)
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)
	SymbolsDynamic [▶ 2009]	Gets the symbols (late bound as dynamic objects)

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetDynamicSymbolsAsync [▶ 2010]	Gets the dynamic symbols asynchronously
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Remarks

Dynamic symbols are late bound symbols who are created and will expand during runtime. These symbols can represent complex user defined type instances like PLC Structures and Arrays created during PLC access and assure type safe to their fields and elements.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.44.1 IDynamicSymbolLoader Properties

The IDynamicSymbolLoader [▶ 2007] type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types. (Inherited from ISymbolLoader [▶ 2200].)
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)

	Name	Description
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)
	Settings [▶ 2202]	Gets or sets the access Method (Inherited from ISymbolLoader [▶ 2200].)
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)
	SymbolsDynamic [▶ 2009]	Gets the symbols (late bound as dynamic objects)

Reference

IDynamicSymbolLoader Interface [▶ 2007]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.44.1.1 IDynamicSymbolLoader.SymbolsDynamic Property

Gets the symbols (late bound as dynamic objects)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDynamicSymbolsCollection SymbolsDynamic { get; }
```

Property Value

Type: IDynamicSymbolsCollection [▶ 2010]
The dynamic symbols.

Reference

IDynamicSymbolLoader Interface [▶ 2007]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.44.2 IDynamicSymbolLoader Methods

The IDynamicSymbolLoader [▶ 2007] type exposes the following members.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetDynamicSymbolsAsync [▶ 2010]	Gets the dynamic symbols asynchronously
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

IDynamicSymbolLoader Interface [▶ 2007]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.44.2.1 IDynamicSymbolLoader.GetDynamicSymbolsAsync Method

Gets the dynamic symbols asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultDynamicSymbols> GetDynamicSymbolsAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultDynamicSymbols [▶ 2350].
Task<ResultDynamicSymbols>.

Reference

IDynamicSymbolLoader Interface [▶ 2007]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.45 IDynamicSymbolsCollection Interface

Interface IDynamicSymbolsContainer Implements the IDynamicMetaObjectProvider

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDynamicSymbolsCollection : IDynamicMetaObjectProvider,
    IEnumerable<ISymbol>, IEnumerable
```

The IDynamicSymbolsCollection type exposes the following members.

Methods

	Name	Description
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ISymbol [▶ 2176]..)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.Dynamic.IDynamicMetaObjectProvider

6.11.45.1 IDynamicSymbolsCollection Methods

The IDynamicSymbolsCollection [▶ 2010] type exposes the following members.

Methods

	Name	Description
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ISymbol [▶ 2176]..)

Reference

IDynamicSymbolsCollection Interface [▶ 2010]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.46 IDynamicValue Interface

Interface IDynamicValue Implements the IDynamicMetaObjectProvider Implements the IValue [▶ 2226] Implements the IStructValue [▶ 2167] Implements the IArrayValue [▶ 1976]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IDynamicValue : IDynamicMetaObjectProvider,
    IValue, IStructValue, IArrayValue
```

The IDynamicValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)

	Name	Description
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TryGetArrayElementValues [▶ 1978]	Returns Array Element values. (Inherited from IArrayValue [▶ 1976].)
	TryGetIndexValue [▶ 1978]	Tries to get the specified Array Element (Inherited from IArrayValue [▶ 1976].)
	TryGetMemberValue [▶ 2169]	Tries to get a property/Member value. (Inherited from IStructValue [▶ 2167].)
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TrySetIndexValue [▶ 1979]	Tries to set the indexed value on Arrays (Inherited from IArrayValue [▶ 1976].)
	TrySetMemberValue [▶ 2170]	Tries to Set a Member/Property Value (Inherited from IStructValue [▶ 2167].)
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.Dynamic.IDynamicMetaObjectProvider

TwinCAT.TypeSystem.IValue [▶ 2226]

TwinCAT.TypeSystem.IStructValue [▶ 2167]

TwinCAT.TypeSystem.IArrayValue [▶ 1976]

6.11.46.1 IDynamicValue Properties

The IDynamicValue [▶ 2011] type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)

	Name	Description
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Reference

IDynamicValue Interface [▶ 2011]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.46.2 IDynamicValue Methods

The IDynamicValue [▶ 2011] type exposes the following members.

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TryGetArrayElementValues [▶ 1978]	Returns Array Element values. (Inherited from IArrayValue [▶ 1976].)
	TryGetIndexValue [▶ 1978]	Tries to get the specified Array Element (Inherited from IArrayValue [▶ 1976].)
	TryGetMemberValue [▶ 2169]	Tries to get a property/Member value. (Inherited from IStructValue [▶ 2167].)
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TrySetIndexValue [▶ 1979]	Tries to set the indexed value on Arrays (Inherited from IArrayValue [▶ 1976].)
	TrySetMemberValue [▶ 2170]	Tries to Set a Member/Property Value (Inherited from IStructValue [▶ 2167].)
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

IDynamicValue Interface [▶ 2011]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47 IEnumType Interface

Common Enum type interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IEnumType : IAliasType,
    IDataType, IBitSize
```

The IEnumType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type (Inherited from IAliasType [▶ 1954].)
	BaseTypeName [▶ 1957]	Gets the BaseType name (Inherited from IAliasType [▶ 1954].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	EnumValues [▶ 2016]	Enumeration specification (if enum)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)

	Name	Description
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Methods

	Name	Description
	Contains [▶ 2017]	Determines whether the enum values contains the specified name
	GetNames [▶ 2018]	Gets the filed names of the IEnumType.T. [▶ 2021]
	GetValues [▶ 2018]	Gets the values of the IEnumType.T. [▶ 2021]
	Parse [▶ 2019]	Parses a value name of the IEnumType.T. [▶ 2021] and returns the value (as base type)
	ToString [▶ 2019]	Returns a String that represents the specified value.
	TryParse(String, IConvertible.) [▶ 2020]	Parses the value from value name.
	TryParse(String, IEnumValue.) [▶ 2021]	Parses the value from value name.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.1 IEnumType Properties

The IEnumType [▶ 2014] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type (Inherited from IAliasType [▶ 1954].)
	BaseTypeName [▶ 1957]	Gets the BaseType name (Inherited from IAliasType [▶ 1954].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)

	Name	Description
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	EnumValues [▶ 2016]	Enumeration specification (if enum)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.1.1 IEnumType.EnumValues Property

Enumeration specification (if enum)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IEnumValueCollection EnumValues { get; }
```

Property Value

Type: IEnumValueCollection [▶ 2031]
The enum specification.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2 IEnumType Methods

The IEnumType [▶ 2014] type exposes the following members.

Methods

	Name	Description
	Contains [▶ 2017]	Determines whether the enum values contains the specified name
	GetNames [▶ 2018]	Gets the filed names of the IEnumType.T. [▶ 2021]
	GetValues [▶ 2018]	Gets the values of the IEnumType.T. [▶ 2021]
	Parse [▶ 2019]	Parses a value name of the IEnumType.T. [▶ 2021] and returns the value (as base type)
	ToString [▶ 2019]	Returns a String that represents the specified value.
	TryParse(String, IConvertible.) [▶ 2020]	Parses the value from value name.
	TryParse(String, IEnumValue.) [▶ 2021]	Parses the value from value name.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.1 IEnumType.Contains Method

Determines whether the enum values contains the specified name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if contains the value, otherwise, false.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.2 IEnumType.GetNames Method

Gets the filed names of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.3 IEnumType.GetValues Method

Gets the values of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IConvertible[] GetValues()
```

Return Value

Type: .IConvertible.
T[].

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.4 IEnumType.Parse Method

Parses a value name of the IEnumType.T. [▶ 2021] and returns the value (as base type)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IConvertible Parse(
    string name
)
```

Parameters

name	Type: System.String The value name.

Return Value

Type: IConvertible
T.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.5 IEnumType.ToString Method

Returns a String that represents the specified value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ToString(
    IConvertible val
)
```

Parameters

val	Type: System.IConvertible The value.

Return Value

Type: String
A String that represents this value.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.47.2.6 IEnumType.TryParse Method

Overload List

	Name	Description
	TryParse(String, IConvertible.) [▶ 2020]	Parses the value from value name.
	TryParse(String, IEnumValue.) [▶ 2021]	Parses the value from value name.

Reference

IEnumType Interface [▶ 2014]

TwinCAT.TypeSystem Namespace [▶ 1622]

IEnumType.TryParse Method (String, IConvertible.)

Parses the value from value name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryParse(
    string name,
    out IConvertible value
)
```

Parameters

name	Type: System.String The value name.
value	Type: System.IConvertible. The value.

Return Value

Type: Boolean
true if value name was found, false otherwise.

Reference

IEnumType Interface [▶ 2014]

TryParse Overload [▶ 2020]

TwinCAT.TypeSystem Namespace [▶ 1622]

IEnumType.TryParse Method (String, IEnumValue.)

Parses the value from value name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryParse(
    string name,
    out IEnumValue value
)
```

Parameters

name	Type: System.String The value name.
value	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The value.

Return Value

Type: Boolean
true if value name was found, false otherwise.

Reference

IEnumType Interface [▶ 2014]

TryParse Overload [▶ 2020]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48 IEnumType.T. Interface

Interface representing an enum type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IEnumType<T> : IAliasType,
    IDataType, IBitSize
where T : IConvertible
```

Type Parameters

T	Base type of the Enum

The IEnumType.T. type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type (Inherited from IAliasType [▶ 1954].)
	BaseTypeName [▶ 1957]	Gets the BaseType name (Inherited from IAliasType [▶ 1954].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	EnumValues [▶ 2024]	Enumeration specification (if enum)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Methods

	Name	Description
	Contains [▶ 2025]	Determines whether the enum values contains the specified name
	GetNames [▶ 2025]	Gets the filed names of the IEnumType.T.
	GetValues [▶ 2026]	Gets the values of the IEnumType.T.
	Parse [▶ 2026]	Parses a name of the IEnumType.T. and returns the value (as base type)

	Name	Description
	ToString [▶ 2027]	Returns a String that represents the specified value.
	TryParse [▶ 2027]	Parses the value from value name.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.1 IEnumType.T. Properties

The IEnumType.T. [▶ 2021] generic type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 1956]	Gets the Base Type (Inherited from IAliasType [▶ 1954].)
	BaseTypeName [▶ 1957]	Gets the BaseType name (Inherited from IAliasType [▶ 1954].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	EnumValues [▶ 2024]	Enumeration specification (if enum)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)

	Name	Description
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.1.1 IEnumType.T..EnumValues Property

Enumeration specification (if enum)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IEnumValueCollection EnumValues { get; }
```

Property Value

Type: IEnumValueCollection [▶ 2031]
The enum specification.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2 IEnumType.T. Methods

The IEnumType.T. [▶ 2021] generic type exposes the following members.

Methods

	Name	Description
	Contains [▶ 2025]	Determines whether the enum values contains the specified name
	GetNames [▶ 2025]	Gets the filed names of the IEnumType.T. [▶ 2021]
	GetValues [▶ 2026]	Gets the values of the IEnumType.T. [▶ 2021]
	Parse [▶ 2026]	Parses a name of the IEnumType.T. [▶ 2021] and returns the value (as base type)
	ToString [▶ 2027]	Returns a String that represents the specified value.
	TryParse [▶ 2027]	Parses the value from value name.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.1 IEnumType.T..Contains Method

Determines whether the enum values contains the specified name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if contains the value, otherwise, false.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.2 IEnumType.T..GetNames Method

Gets the filed names of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.3 IEnumType.T..GetValues Method

Gets the values of the IEnumType.T. [▶ 2021]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T[] GetValues()
```

Return Value

Type: .T [▶ 2021].
T[].

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.4 IEnumType.T..Parse Method

Parses a name of the IEnumType.T. [▶ 2021] and returns the value (as base type)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 2021]
T.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.5 IEnumType.T..ToString Method

Returns a String that represents the specified value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ToString(
    T val
)
```

Parameters

val	Type: T [▶ 2021] The value.

Return Value

Type: String
A String that represents this value.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.48.2.6 IEnumType.T..TryParse Method

Parses the value from value name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryParse(
    string name,
    out T value
)
```

Parameters

name	Type: System.String The value name.
value	Type: T [▶ 2021]. The value.

Return Value

Type: Boolean
true if value name was found, false otherwise.

Reference

IEnumType.T. Interface [▶ 2021]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49 IEnumValue Interface

Generic interface for EnumValues

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IEnumValue
```

The IEnumValue type exposes the following members.

Properties

	Name	Description
	ManagedBaseType [▶ 2029]	Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)
	Name [▶ 2029]	Gets the name of the Enumeration Value (value as string)
	Primitive [▶ 2029]	Gets the (Primitive, BaseType) Value of the enumeration as object
	RawValue [▶ 2030]	Gets the raw value of the enumeration (as byte array)
	Size [▶ 2030]	Gets the size of the Enum value (in bytes)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1 IEnumValue Properties

The IEnumValue [▶ 2028] type exposes the following members.

Properties

	Name	Description
	ManagedBaseType [▶ 2029]	Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)
	Name [▶ 2029]	Gets the name of the Enumeration Value (value as string)
	Primitive [▶ 2029]	Gets the (Primitive, BaseType) Value of the enumeration as object
	RawValue [▶ 2030]	Gets the raw value of the enumeration (as byte array)
	Size [▶ 2030]	Gets the size of the Enum value (in bytes)

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1.1 IEnumValue.ManagedBaseType Property

Gets the enumeration base type (sint,byte,short,ushort,int,uint,Int64,UInt64 supported)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Type ManagedBaseType { get; }
```

Property Value

Type: Type
The type of the base.

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1.2 IEnumValue.Name Property

Gets the name of the Enumeration Value (value as string)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1.3 IEnumValue.Primitive Property

Gets the (Primitive, BaseType) Value of the enumeration as object

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object Primitive { get; }
```

Property Value

Type: Object
The object value.

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1.4 IEnumValue.RawValue Property

Gets the raw value of the enumeration (as byte array)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
byte[] RawValue { get; }
```

Property Value

Type: .Byte.
The raw value.

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.49.1.5 IEnumValue.Size Property

Gets the size of the Enum value (in bytes)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Size { get; }
```

Property Value

Type: Int32
The size.

Reference

IEnumValue Interface [▶ 2028]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.50 IEnumValueCollection Interface

Interface IEnumValueCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IEnumValueCollection : IEnumValueCollection<IEnumValue, IConvertible>,
    ICollection<IEnumValue>, IEnumerable<IEnumValue>, IEnumerable
```

The IEnumValueCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Item [▶ 2035]	Gets or sets the element at the specified index. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Contains(String) [▶ 2036]	Determines whether [contains] [the specified name]. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IEnumValue [▶ 2028]..)
	GetNames [▶ 2037]	Gets the Value Names. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)

	Name	Description
	GetValues [▶ 2037]	Gets the values. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Parse [▶ 2038]	Parses the specified name. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	TryParse(String, TValue.) [▶ 2039]	Tries to parse the string value of the Enum. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.50.1 IEnumValueCollection Properties

The IEnumValueCollection [▶ 2031] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Item [▶ 2035]	Gets or sets the element at the specified index. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)

Reference

IEnumValueCollection Interface [▶ 2031]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.50.2 IEnumValueCollection Methods

The IEnumValueCollection [▶ 2031] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	Contains(String) [▶ 2036]	Determines whether [contains] [the specified name]. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IEnumValue [▶ 2028]..)

	Name	Description
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IEnumValue [▶ 2028]..)
	GetNames [▶ 2037]	Gets the Value Names. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	GetValues [▶ 2037]	Gets the values. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Parse [▶ 2038]	Parses the specified name. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IEnumValue [▶ 2028]..)
	TryParse(String, TValue.) [▶ 2039]	Tries to parse the string value of the Enum. (Inherited from IEnumValueCollection.TEnumValue, TValue. [▶ 2033].)

Reference

IEnumValueCollection Interface [▶ 2031]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51 IEnumValueCollection.TEnumValue, TValue. Interface

Interface for collections of IEnumValues [▶ 2028]. Implements the ICollection.T.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IEnumValueCollection<TEnumValue, TValue> : ICollection<TEnumValue>,
    IEnumerable<TEnumValue>, IEnumerable
where TEnumValue : class, IEnumValue
where TValue : IConvertible
```

Type Parameters

TEnumValue	TValue

The IEnumValueCollection.TEnumValue, TValue. type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.TEnumValue..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.TEnumValue..)
	Item [▶ 2035]	Gets or sets the element at the specified index.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.TEnumValue..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.TEnumValue..)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.TEnumValue..)
	Contains(String) [▶ 2036]	Determines whether [contains] [the specified name].
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.TEnumValue..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.TEnumValue..)
	GetNames [▶ 2037]	Gets the Value Names.
	GetValues [▶ 2037]	Gets the values.
	Parse [▶ 2038]	Parses the specified name.
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.TEnumValue..)
	TryParse(String, TEnumValue.) [▶ 2039]	Tries to parse the string value of the Enum.
	TryParse(String, TValue.) [▶ 2039]	Tries to parse the string value of the Enum.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

System.Collections.Generic.ICollection.T.

6.11.51.1 IEnumValueCollection.TEnumValue, TValue. Properties

The IEnumValueCollection.TEnumValue, TValue. [▶ 2033] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	Item [▶ 2035]	Gets or sets the element at the specified index.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.1.1 IEnumValueCollection.TEnumValue, TValue..Item Property

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TValue this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name of the value

Return Value

Type: TValue [▶ 2033]
EnumValue<T>.

Exceptions

Exception	Condition
NotImplementedException	
NotImplementedException	

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2 IEnumValueCollection.TEnumValue, TValue. Methods

The IEnumValueCollection.TEnumValue, TValue. [▶ 2033] generic type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	Contains(String) [▶ 2036]	Determines whether [contains] [the specified name].

	Name	Description
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.TEnumValue [▶ 2033]..)
	GetNames [▶ 2037]	Gets the Value Names.
	GetValues [▶ 2037]	Gets the values.
	Parse [▶ 2038]	Parses the specified name.
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	TryParse(String, TEnumValue.) [▶ 2039]	Tries to parse the string value of the Enum.
	TryParse(String, TValue.) [▶ 2039]	Tries to parse the string value of the Enum.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2.1 IEnumValueCollection.TEnumValue, TValue..Contains Method

Overload List

	Name	Description
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.TEnumValue [▶ 2033]..)
	Contains(String) [▶ 2036]	Determines whether [contains] [the specified name].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

IEnumValueCollection.TEnumValue, TValue..Contains Method (String)

Determines whether [contains] [the specified name].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string value
)
```

Parameters

value	Type: System.String Value

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

Contains Overload [▶ 2036]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2.2 IEnumValueCollection.TEnumValue, TValue..GetNames Method

Gets the Value Names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2.3 IEnumValueCollection.TEnumValue, TValue..GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TValue[] GetValues()
```

Return Value

Type: .TValue [▶ 2033].
T[].

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2.4 IEnumValueCollection.TEnumValue, TValue..Parse Method

Parses the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TValue Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: TValue [▶ 2033]
T.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.51.2.5 IEnumValueCollection.TEnumValue, TValue..TryParse Method

Overload List

	Name	Description
	TryParse(String, TEnumValue.) [▶ 2039]	Tries to parse the string value of the Enum.
	TryParse(String, TValue.) [▶ 2039]	Tries to parse the string value of the Enum.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TwinCAT.TypeSystem Namespace [▶ 1622]

IEnumValueCollection.TEnumValue, TValue..TryParse Method (String, TEnumValue.)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryParse(
    string strValue,
    out TEnumValue value
)
```

Parameters

strValue	Type: System.String The Value in string representation.
value	Type: TEnumValue [▶ 2033]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TryParse Overload [▶ 2038]

TwinCAT.TypeSystem Namespace [▶ 1622]

IEnumValueCollection.TEnumValue, TValue..TryParse Method (String, TValue.)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryParse(
    string strValue,
    out TValue value
)
```

Parameters

strValue	Type: System.String The Value in string representation.
value	Type: TValue [▶ 2033]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

IEnumValueCollection.TEnumValue, TValue. Interface [▶ 2033]

TryParse Overload [▶ 2038]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.52 IField Interface

Specifies a single field/member of a Struct DataType [▶ 2162].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IField : IAttributedInstance, IInstance,
    IBitSize
```

The IField type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	ParentType [▶ 2042]	Gets the Parent Struct/Union of this IField.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.52.1 IField Properties

The IField [▶ 2040] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)

	Name	Description
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	ParentType [▶ 2042]	Gets the Parent Struct/Union of this IField [▶ 2040].
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IField Interface [▶ 2040]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.52.1.1 IField.ParentType Property

Gets the Parent Struct/Union of this IField [▶ 2040].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ParentType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the parent.

Reference

IField Interface [▶ 2040]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.53 IFieldCollection Interface

Interface IFieldCollection Implements the IInstanceCollection.T. [▶ 2057]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IFieldCollection : IInstanceCollection<IField>,
    IList<IField>, ICollection<IField>, IEnumerable<IField>,
    IEnumerable
```

The IFieldCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IField [▶ 2040]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IField [▶ 2040]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IField [▶ 2040]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IField [▶ 2040]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IField [▶ 2040]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IField [▶ 2040]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IField [▶ 2040]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IField [▶ 2040]..)

	Name	Description
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetMember [▶ 2045]	Gets the Field/Member with the specified name from the collection.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IInstanceCollection.T. [▶ 2057]

6.11.53.1 IFieldCollection Properties

The IFieldCollection [▶ 2042] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IField [▶ 2040]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IField [▶ 2040]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

IFieldCollection Interface [▶ 2042]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.53.2 IFieldCollection Methods

The IFieldCollection [▶ 2042] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)

	Name	Description
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IField [▶ 2040]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IField [▶ 2040]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IField [▶ 2040]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IField [▶ 2040]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IField [▶ 2040]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IField [▶ 2040]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IField [▶ 2040]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetMember [▶ 2045]	Gets the Field/Member with the specified name from the collection.

Reference

IFieldCollection Interface [▶ 2042]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.53.2.1 IFieldCollection.TryGetMember Method

Gets the Field/Member with the specified name from the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetMember(
    string fieldName,
    out IField symbol
)
```

Parameters

fieldName	Type: System.String Name of the field/member.
symbol	Type: TwinCAT.TypeSystem.IField [▶ 2040]. The symbol with the specified name.

Return Value

Type: Boolean
true if found, false otherwise.

Reference

IFieldCollection Interface [▶ 2042]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.54 IGenericTypeMarshaler Interface

Interface IGenericTypeMarshaller Implements the ITypeMarshaler [▶ 2216]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IGenericTypeMarshaler : ITypeMarshaler
```

The IGenericTypeMarshaler type exposes the following members.

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [▶ 2216].)
	Marshal [▶ 2219]	(Inherited from ITypeMarshaler [▶ 2216].)
	MarshalSize [▶ 2219]	Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal.T.(ReadOnlySpan, Void, Byte) [▶ 2048]	

Remarks

The IGenericTypeMarshaler extends the ITypeMarshaler [▶ 2216] by generic methods.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.ITypeMarshaler [▶ 2216]

6.11.54.1 IGenericTypeMarshaler Methods

The IGenericTypeMarshaler [▶ 2046] type exposes the following members.

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value (Inherited from ITypeMarshaler [▶ 2216].)
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type. (Inherited from ITypeMarshaler [▶ 2216].)
	Marshal [▶ 2219]	(Inherited from ITypeMarshaler [▶ 2216].)
	MarshalSize [▶ 2219]	Gets the byte size of the value when marshalled. (Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)
	Unmarshal.T. (ReadOnlySpan, Void, Byte) [▶ 2048]	

Reference

IGenericTypeMarshaler Interface [▶ 2046]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.54.1.1 IGenericTypeMarshaler.Unmarshal Method

Overload List

	Name	Description
	Unmarshal.T. (ReadOnlySpan, Void, Byte) [▶ 2048]	
	Unmarshal(Type, ReadOnlySpan, Void, Byte) [▶ 2220]	(Inherited from ITypeMarshaler [▶ 2216].)

Reference

IGenericTypeMarshaler Interface [▶ 2046]

TwinCAT.TypeSystem Namespace [▶ 1622]

IGenericTypeMarshaler.Unmarshal.T. Method (ReadOnlySpan`1, Void, Byte)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Unmarshal<T>(
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters

source	Type: ReadOnlySpan
encoding	Type: System.Void
value	Type: System.Byte

Type Parameters

T

Return Value

Type: Int32

Reference

IGenericTypeMarshaler Interface [▶ 2046]

Unmarshal Overload [▶ 2047]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.55 IHierarchicalSymbol Interface

Bindable Symbol interface (for internal use only)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IHierarchicalSymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IHierarchicalSymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	SetParent [▶ 2051]	Sets the parent of the Symbol

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.55.1 IHierarchicalSymbol Properties

The IHierarchicalSymbol [▶ 2048] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)

	Name	Description
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IHierarchicalSymbol Interface [▶ 2048]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.55.2 IHierarchicalSymbol Methods

The IHierarchicalSymbol [▶ 2048] type exposes the following members.

Methods

	Name	Description
	SetParent [▶ 2051]	Sets the parent of the Symbol

Reference

IHierarchicalSymbol Interface [▶ 2048]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.55.2.1 IHierarchicalSymbol.SetParent Method

Sets the parent of the Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void SetParent(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

IHierarchicalSymbol Interface [▶ 2048]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56 IInstance Interface

Interface specifying instance objects.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IInstance : IBitSize
```

The IInstance type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance.
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance represents a Pointer type (Pointer TO)
	IsReference [▶ 2056]	Indicates that the IInstance represents a Reference type (REFERENCE TO)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance is static.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1 IInstance Properties

The IInstance [▶ 2052] type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052]
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.))
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.1 IInstance.Comment Property

Gets the comment of the IInstance [▶ 2052]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Comment { get; }
```

Property Value

Type: String
The comment.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.2 IInstance.DataType Property

Gets the IDataType [▶ 1986] of the IInstance [▶ 2052].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType DataType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Remarks

The DataType can be unresolved in rare circumstances and therefore could have value **null**.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.3 IInstance.InstanceName Property

Gets the name of the instance (without periods (.)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string InstanceName { get; }
```

Property Value

Type: String
The name of the instance.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.4 IInstance.InstancePath Property

Gets the relative / absolute access path to the instance (with periods (.))

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Remarks

If this path is relative or absolute depends on the context. IMember [▶ 2065] are using relative paths, ISymbol [▶ 2176]s are using absolute ones.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.5 IInstance.IsPointer Property

Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPointer { get; }
```

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.6 IInstance.IsReference Property

Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsReference { get; }
```

Property Value

Type: Boolean
true if is ReferenceTo, otherwise false.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.7 IInstance.IsStatic Property

Gets a value indicating whether this IInstance [▶ 2052] is static.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsStatic { get; }
```

Property Value

Type: Boolean
true if this instance is static; otherwise, false.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.56.1.8 IInstance.TypeName Property

Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string TypeName { get; }
```

Property Value

Type: String
The name of the type.

Reference

IInstance Interface [▶ 2052]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57 IInstanceCollection.T. Interface

Generic InstanceCollection interface.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IInstanceCollection<T> : IList<T>,
    ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, IInstance
```

Type Parameters

T

The IInstanceCollection.T. type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.T..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path.
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075].

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T..)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path.
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name.
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name.
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.T..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.T..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.T..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance.
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.1 IInstanceCollection.T. Properties

The IInstanceCollection.T. [▶ 2057] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T [▶ 2057]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T [▶ 2057]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.T [▶ 2057]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path.

	Name	Description
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075].

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.1.1 IInstanceCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.T [▶ 2057]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

IInstanceCollection.T..Item Property (String)

Gets the IInstance [▶ 2052] with the specified instance path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T this[
    string instancePath
] { get; }
```

Parameters

instancePath	Type: System.String

Property Value

Type: T [▶ 2057]

Reference

IInstanceCollection.T. Interface [▶ 2057]

Item Overload [▶ 2059]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.1.2 IInstanceCollection.T..Mode Property

Gets the InstanceCollectionMode [▶ 2075].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
InstanceCollectionMode Mode { get; }
```

Property Value

Type: InstanceCollectionMode [▶ 2075]
The mode.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2 IInstanceCollection.T. Methods

The IInstanceCollection.T. [▶ 2057] generic type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T [▶ 2057]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T [▶ 2057]..)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T [▶ 2057]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path.
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name.
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T [▶ 2057]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T [▶ 2057]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name.
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.T [▶ 2057]..)

	Name	Description
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.T [▶ 2057]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T [▶ 2057]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.T [▶ 2057]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance.
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.1 IInstanceCollection.T..Contains Method

Overload List

	Name	Description
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T [▶ 2057]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

IInstanceCollection.T..Contains Method (String)

Determines whether this collection contains an instance with the specified instance path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string instancePath
)
```

Parameters

instancePath	Type: System.String The instance path.

Return Value

Type: Boolean
true if this collection contains the specified instance path; otherwise, false.

Reference

IInstanceCollection.T. Interface [▶ 2057]

Contains Overload [▶ 2061]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.2 IInstanceCollection.T..ContainsName Method

Determines whether this collection contains an instance with the specified instance name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool ContainsName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: Boolean
true if this collection contains the specified instance path; otherwise, false.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.3 IInstanceCollection.T..GetInstance Method

Gets the IInstance [▶ 2052]by instance path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T GetInstance(
    string instancePath
)
```

Parameters

instancePath	Type: System.String The instance path.

Return Value

Type: T [▶ 2057]
T.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.4 IInstanceCollection.T..GetInstanceByName Method

Gets the IInstance [▶ 2052] by instance name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IList<T> GetInstanceByName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: IList.T [▶ 2057].
IList<T>.

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.5 IInstanceCollection.T..TryGetInstance Method

Tries to get the specified instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetInstance(
    string instancePath,
    out T symbol
)
```

Parameters

instancePath	Type: System.String The instance path.
symbol	Type: T [▶ 2057]. The symbol.

Return Value

Type: Boolean
true if the IInstance [▶ 2052] is found; otherwise, false

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.57.2.6 IInstanceCollection.T..TryGetInstanceByName Method

Tries to get the specified instance by name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetInstanceByName(
    string instanceName,
    out IList<T> symbols
)
```

Parameters

instanceName	Type: System.String Name of the instance.
symbols	Type: System.Collections.Generic.IList.T [▶ 2057].. The found symbols.

Return Value

Type: Boolean
true if the IInstance [▶ 2052] is found; otherwise, false

Reference

IInstanceCollection.T. Interface [▶ 2057]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.58 IMember Interface

Specifies a single field/member of a Struct DataType [▶ 2162].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IMember : IField,
    IAttributedInstance, IInstance, IBitSize
```

The IMember type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitOffset [▶ 2067]	Gets the bit offset.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteOffset [▶ 2067]	Gets the byte offset.
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Offset [▶ 2068]	Gets the offset of the IMember within the parent IStructType [▶ 2162] in bits or bytes dependent on IsBitType.

	Name	Description
	ParentType [▶ 2042]	Gets the Parent Struct/Union of this IField [▶ 2040]. (Inherited from IField [▶ 2040].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.58.1 IMember Properties

The IMember [▶ 2065] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitOffset [▶ 2067]	Gets the bit offset.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteOffset [▶ 2067]	Gets the byte offset.
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)

	Name	Description
	Offset [▶ 2068]	Gets the offset of the IMember [▶ 2065] within the parent IStructType [▶ 2162] in bits or bytes dependent on IsBitType.
	ParentType [▶ 2042]	Gets the Parent Struct/Union of this IField [▶ 2040]. (Inherited from IField [▶ 2040].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IMember Interface [▶ 2065]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.58.1.1 IMember.BitOffset Property

Gets the bit offset.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int BitOffset { get; }
```

Property Value

Type: Int32
The bit offset.

Reference

IMember Interface [▶ 2065]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.58.1.2 IMember.ByteOffset Property

Gets the byte offset.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ByteOffset { get; }
```

Property Value

Type: Int32
The byte offset.

Reference

IMember Interface [▶ 2065]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.58.1.3 IMember.Offset Property

Gets the offset of the IMember [▶ 2065] within the parent IStructType [▶ 2162] in bits or bytes dependent on IsBitType.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Offset { get; }
```

Property Value

Type: Int32
The bit offset.

Reference

IMember Interface [▶ 2065]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.59 IMemberCollection Interface

Interface IMemberCollection Implements the IInstanceCollection.T. [▶ 2057]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IMemberCollection : IInstanceCollection<IMember>,
    IList<IMember>, ICollection<IMember>, IEnumerable<IMember>,
    IEnumerable
```

The IMemberCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IMember [▶ 2065]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IMember [▶ 2065]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IMember [▶ 2065]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IMember [▶ 2065]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IMember [▶ 2065]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetMember [▶ 2071]	Gets the member with the specified name from the collection.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IInstanceCollection.T. [▶ 2057]

6.11.59.1 IMemberCollection Properties

The IMemberCollection [▶ 2068] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IMember [▶ 2065]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

IMemberCollection Interface [▶ 2068]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.59.2 IMemberCollection Methods

The IMemberCollection [▶ 2068] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IMember [▶ 2065]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IMember [▶ 2065]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IMember [▶ 2065]..)

	Name	Description
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IMember [▶ 2065]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IMember [▶ 2065]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IMember [▶ 2065]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetMember [▶ 2071]	Gets the member with the specified name from the collection.

Reference

IMemberCollection Interface [▶ 2068]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.59.2.1 IMemberCollection.TryGetMember Method

Gets the member with the specified name from the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetMember(
    string memberName,
    out IMember symbol
)
```

Parameters

memberName	Type: System.String Name of the member.
symbol	Type: TwinCAT.TypeSystem.IMember [▶ 2065]. The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Reference

IMemberCollection Interface [▶ 2068]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.60 INamespaceCollection Interface

Interface INamespaceCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface INamespaceCollection : INamespaceCollection<IDataType>,
    ICollection<INamespace<IDataType>>, IEnumerable<INamespace<IDataType>>,
    IEnumerable
```

The INamespaceCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.60.1 INamespaceCollection Properties

The INamespaceCollection [▶ 2072] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)

Reference

INamespaceCollection Interface [▶ 2072]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.60.2 INamespaceCollection Methods

The INamespaceCollection [▶ 2072] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.INamespace [▶ 2458].IDataType [▶ 1986]...)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].IDataType [▶ 1986]...)

Reference

INamespaceCollection Interface [▶ 2072]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.61 INamespaceCollection.T. Interface

Interface INamespaceCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface INamespaceCollection<T> : ICollection<INamespace<T>>,
    IEnumerable<INamespace<T>>, IEnumerable
where T : class, IDataType
```

Type Parameters

T

The INamespaceCollection.T. type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T...)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.INamespace [▶ 2458].T...)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T...)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T...)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.INamespace [▶ 2458].T...)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.INamespace [▶ 2458].T...)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.INamespace [▶ 2458].T...)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T...)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.61.1 INamespaceCollection.T. Properties

The INamespaceCollection.T. [▶ 2073] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)

Reference

INamespaceCollection.T. Interface [▶ 2073]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.61.2 INamespaceCollection.T. Methods

The INamespaceCollection.T. [▶ 2073] generic type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.INamespace [▶ 2458].T [▶ 2073]...)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.INamespace [▶ 2458].T [▶ 2073]...)

Reference

INamespaceCollection.T. Interface [▶ 2073]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.62 InstanceCollectionMode Enumeration

Enum InstanceCollectionMode

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum InstanceCollectionMode
```

Members

	Member name	Value	Description
	Names	0	InstanceCollection{T} is organized with InstanceNames instead of Instance Paths
	Path	1	InstanceCollection{T} is organized with InstancePaths in a flat list

	Member name	Value	Description
	PathHierarchy	2	InstanceCollection{T} is organized with InstancePaths in a Hierarchy (Only Root objects appearing)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.63 InsufficientAccessRightsException Class

Insufficient rights for access

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]
        TwinCAT.TypeSystem.InsufficientAccessRightsException

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class InsufficientAccessRightsException : SymbolException
```

The InsufficientAccessRightsException type exposes the following members.

Constructors

	Name	Description
	InsufficientAccessRightsException [▶ 2077]	Initializes a new instance of the InsufficientAccessRightsException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)

	Name	Description
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.63.1 InsufficientAccessRightsException Constructor

Initializes a new instance of the InsufficientAccessRightsException [▶ 2076] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public InsufficientAccessRightsException(
    IValueSymbol symbol,
    SymbolAccessRights requested
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.IValueSymbol [▶ 2254] The symbol.
requested	Type: TwinCAT.TypeSystem.SymbolAccessRights [▶ 2396] The requested.

Reference

InsufficientAccessRightsException Class [▶ 2076]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.63.2 InsufficientAccessRightsException Properties

The InsufficientAccessRightsException [▶ 2076] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

InsufficientAccessRightsException Class [▶ 2076]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.63.3 InsufficientAccessRightsException Methods

The InsufficientAccessRightsException [▶ 2076] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)

	Name	Description
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

InsufficientAccessRightsException Class [▶ 2076]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.64 IOversamplingArrayInstance Interface

Interface IOversamplingArrayInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IOversamplingArrayInstance : IArrayInstance,
    ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IOversamplingArrayInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	Dimensions [▶ 1967]	Gets the dimensions as read only collection. (Inherited from IArrayInstance [▶ 1964].)
	Elements [▶ 1968]	Gets the contained Array Elements as read only collection. (Inherited from IArrayInstance [▶ 1964].)
	ElementType [▶ 1968]	Gets the type of the contained elements. (Inherited from IArrayInstance [▶ 1964].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Item [▶ 1969]	Gets the ISymbol [▶ 2176] with the specified indices. (Inherited from IArrayInstance [▶ 1964].)
	OversamplingElement [▶ 2082]	Gets the oversampling element.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	TryGetElement(IList.., Int32.., ISymbol.) [▶ 1970]	Tries to get the array element with the specified indices (jagged array support). (Inherited from IArrayInstance [▶ 1964].)
	TryGetElement(.Int32., ISymbol.) [▶ 1971]	Tries to get the array element with specified indices (only first level on jagged arrays) (Inherited from IArrayInstance [▶ 1964].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IArrayInstance [▶ 1964]

6.11.64.1 IOversamplingArrayInstance Properties

The IOversamplingArrayInstance [▶ 2079] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	Dimensions [▶ 1967]	Gets the dimensions as read only collection. (Inherited from IArrayInstance [▶ 1964].)
	Elements [▶ 1968]	Gets the contained Array Elements as read only collection. (Inherited from IArrayInstance [▶ 1964].)
	ElementType [▶ 1968]	Gets the type of the contained elements. (Inherited from IArrayInstance [▶ 1964].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)

	Name	Description
	Item [▶ 1969]	Gets the ISymbol [▶ 2176] with the specified indices. (Inherited from IArrayInstance [▶ 1964].)
	OversamplingElement [▶ 2082]	Gets the oversampling element.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IOversamplingArrayInstance Interface [▶ 2079]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.64.1.1 IOversamplingArrayInstance.OversamplingElement Property

Gets the oversampling element.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol OversamplingElement { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The oversampling element.

Reference

IOversamplingArrayInstance Interface [▶ 2079]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.64.2 IOversamplingArrayInstance Methods

Methods

	Name	Description
	TryGetElement(IList.., Int32.., ISymbol.) [▶ 1970]	Tries to get the array element with the specified indices (jagged array support). (Inherited from IArrayInstance [▶ 1964].)
	TryGetElement(.Int32., ISymbol.) [▶ 1971]	Tries to get the array element with specified indices (only first level on jagged arrays) (Inherited from IArrayInstance [▶ 1964].)

Reference

IOversamplingArrayInstance Interface [▶ 2079]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.65 IPointerInstance Interface

Interface representing an instance of an IPointerType [▶ 2086]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IPointerInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IPointerInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)

	Name	Description
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Reference [▶ 2086]	Gets the resolved reference of Pointer / Reference
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.65.1 IPointerInstance Properties

The IPointerInstance [▶ 2083] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)

	Name	Description
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Reference [▶ 2086]	Gets the resolved reference of Pointer / Reference
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IPointerInstance Interface [▶ 2083]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.65.1.1 IPointerInstance.Reference Property

Gets the resolved reference of Pointer / Reference

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol Reference { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The reference symbol or NULL if PVOID Pointer.

Reference

IPointerInstance Interface [▶ 2083]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.66 IPointerType Interface

Interface representing a pointer type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IPointerType : IDataType,
    IBitSize
```

The IPointerType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)

	Name	Description
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	ReferencedType [▶ 2088]	Gets the the referenced type.
	ReferenceTypeName [▶ 2089]	Gets the name of the referenced datatype
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.66.1 IPointerType Properties

The IPointerType [▶ 2086] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)

	Name	Description
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	ReferencedType [▶ 2088]	Gets the the referenced type.
	ReferenceTypeName [▶ 2089]	Gets the name of the referenced datatype
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IPointerType Interface [▶ 2086]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.66.1.1 IPointerType.ReferencedType Property

Gets the the referenced type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The datatype of the reference or **null** if not resolved.

Reference

IPointerType Interface [▶ 2086]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.66.1.2 IPointerType.ReferenceTypeName Property

Gets the name of the referenced datatype

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ReferenceTypeName { get; }
```

Property Value

Type: String
The name of the reference dattype.

Reference

IPointerType Interface [▶ 2086]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.67 IPrimitiveType Interface

Interface IPrimitiveType

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IPrimitiveType : IDataType,
    IBitSize
```

The IPrimitiveType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)

	Name	Description
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	PrimitiveFlags [▶ 2091]	Indicates types of different PrimitiveTypes with flags.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IDataType [▶ 1986]

6.11.67.1 IPrimitiveType Properties

The IPrimitiveType [▶ 2089] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)

	Name	Description
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	PrimitiveFlags [▶ 2091]	Indicates types of different PrimitiveTypes with flags.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IPrimitiveType Interface [▶ 2089]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.67.1.1 IPrimitiveType.PrimitiveFlags Property

Indicates types of different PrimitiveTypes with flags.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
PrimitiveTypeFlags PrimitiveFlags { get; }
```

Property Value

Type: PrimitiveTypeFlags [▶ 2289]

Reference

IPrimitiveType Interface [▶ 2089]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.68 IProcessImageAddress Interface

Interface describing a Process Image Address

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IProcessImageAddress : IBitSize
```

The IProcessImageAddress type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	IndexGroup [▶ 2093]	Gets the index group of the Symbol
	IndexOffset [▶ 2093]	Gets the index offset of the Symbol
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsVirtual [▶ 2094]	Gets a value indicating whether this instance is virtual.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.68.1 IProcessImageAddress Properties

The IProcessImageAddress [▶ 2092] type exposes the following members.

Properties

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IndexGroup [▶ 2093]	Gets the index group of the Symbol
	IndexOffset [▶ 2093]	Gets the index offset of the Symbol
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsVirtual [▶ 2094]	Gets a value indicating whether this instance is virtual.
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IProcessImageAddress Interface [▶ 2092]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.68.1.1 IProcessImageAddress.IndexGroup Property

Gets the index group of the Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint IndexGroup { get; }
```

Property Value

Type: UInt32
The index group.

Reference

IProcessImageAddress Interface [▶ 2092]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.68.1.2 IProcessImageAddress.IndexOffset Property

Gets the index offset of the Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
uint IndexOffset { get; }
```

Property Value

Type: UInt32
The index offset.

Reference

IProcessImageAddress Interface [▶ 2092]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.68.1.3 IProcessImageAddress.IsVirtual Property

Gets a value indicating whether this instance is virtual.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsVirtual { get; }
```

Property Value

Type: Boolean
true if this instance is virtual; otherwise, false.

Remarks

Virtual symbols are are only organizational elements within the Symbols Hierarchy and cannot be accessed seperately by IndexGroup/IndexOffset, Value Read/Writes, notifications or handles.

Reference

IProcessImageAddress Interface [▶ 2092]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69 IReferenceInstance Interface

Interface representing an instance of an IReferenceType [▶ 2099]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IReferenceInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IReferenceInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	ReferencedType [▶ 2097]	Gets the referenced type

	Name	Description
	ResolvedByteSize [▶ 2098]	Get the ByteSize of the (completely) resolved Symbol
	ResolvedCategory [▶ 2098]	Gets the Category of the (completely) resolved Symbol.
	ResolvedType [▶ 2099]	Gets the (completely) resolved type
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69.1 IReferenceInstance Properties

The IReferenceInstance [▶ 2094] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	ReferencedType [▶ 2097]	Gets the referenced type
	ResolvedByteSize [▶ 2098]	Get the ByteSize of the (completely) resolved Symbol
	ResolvedCategory [▶ 2098]	Gets the Category of the (completely) resolved Symbol.
	ResolvedType [▶ 2099]	Gets the (completely) resolved type
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IReferenceInstance Interface [▶ 2094]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69.1.1 IReferenceInstance.ReferencedType Property

Gets the referenced type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the referenced type

Remarks

This is no complete resolvement, only the next level. The referenced type can be a reference again.

Reference

IReferenceInstance Interface [▶ 2094]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69.1.2 IReferenceInstance.ResolvedByteSize Property

Get the ByteSize of the (completely) resolved Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ResolvedByteSize { get; }
```

Property Value

Type: Int32
The byte size of the resolved type.

Reference

IReferenceInstance Interface [▶ 2094]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69.1.3 IReferenceInstance.ResolvedCategory Property

Gets the Category of the (completely) resolved Symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DataTypeCategory ResolvedCategory { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The resolved category.

Reference

IReferenceInstance Interface [▶ 2094]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.69.1.4 IReferenceInstance.ResolvedType Property

Gets the (completely) resolved type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ResolvedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the resolved symbol

Reference

IReferenceInstance Interface [▶ 2094]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70 IReferenceType Interface

Interface representing a reference/pointer type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IReferenceType : IDataType,
    IBitSize
```

The IReferenceType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)

	Name	Description
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	ReferencedType [▶ 2102]	Gets the the referenced type.
	ReferencedTypeName [▶ 2102]	Gets the name of the referenced type.
	ResolvedByteSize [▶ 2102]	Get the ByteSize of the (completely) resolved Symbol
	ResolvedCategory [▶ 2103]	Gets the Category of the (completely) resolved Symbol.
	ResolvedType [▶ 2103]	Gets the (completely) resolved type
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1 IReferenceType Properties

The IReferenceType [▶ 2099] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	ReferencedType [▶ 2102]	Gets the the referenced type.
	ReferencedTypeName [▶ 2102]	Gets the name of the referenced type.
	ResolvedByteSize [▶ 2102]	Get the ByteSize of the (completely) resolved Symbol
	ResolvedCategory [▶ 2103]	Gets the Category of the (completely) resolved Symbol.
	ResolvedType [▶ 2103]	Gets the (completely) resolved type
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1.1 IReferenceType.ReferencedType Property

Gets the the referenced type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ReferencedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The data type of the referenced type or **null** if not resolved.

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1.2 IReferenceType.ReferencedTypeName Property

Gets the name of the referenced type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ReferencedTypeName { get; }
```

Property Value

Type: String
The name of the referenced type.

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1.3 IReferenceType.ResolvedByteSize Property

Get the ByteSize of the (completely) resolved Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ResolvedByteSize { get; }
```

Property Value

Type: Int32
The size of the resolved byte.

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1.4 IReferenceType.ResolvedCategory Property

Gets the Category of the (completely) resolved Symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DataTypeCategory ResolvedCategory { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The resolved category.

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.70.1.5 IReferenceType.ResolvedType Property

Gets the (completely) resolved type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType ResolvedType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the resolved symbol

Reference

IReferenceType Interface [▶ 2099]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71 IRpcCallableInstance Interface

Interface for an RPC callable PLC Method (Remote procedure call)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcCallableInstance
```

The IRpcCallableInstance type exposes the following members.

Properties

	Name	Description
	RpcMethods [▶ 2105]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Methods

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 2107]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]	Invokes the specified RPC Method asynchronously

	Name	Description
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]	Invokes the specified RPC Method asynchronously
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]	Invokes the the specified RpcMethod of the IRpcCallableInstance.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.1 IRpcCallableInstance Properties

The IRpcCallableInstance [▶ 2104] type exposes the following members.

Properties

	Name	Description
	RpcMethods [▶ 2105]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.1.1 IRpcCallableInstance.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodCollection RpcMethods { get; }
```

Property Value

Type: IRpcMethodCollection [▶ 2127]
The methods.

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.2 IRpcCallableInstance Methods

Methods

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 2107]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]	Invokes the specified RPC Method
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]	Invokes the specified RPC Method asynchronously
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]	Invokes the specified RPC Method

	Name	Description
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.2.1 IRpcCallableInstance.InvokeRpcMethod Method

Overload List

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 2107]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]	Invokes the specified RPC Method
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]	Invokes the specified RPC Method

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.InvokeRpcMethod Method (String, .Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL

Return Value

Type: Object
The return value of the Method (as object).

Remarks

This method only supports primitive data types as inParameters. Any available outparameters will be ignored. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});
```

```
        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

InvokeRpcMethod Overload [▶ 2107]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.InvokeRpcMethod Method (String, .Object., .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The input parameters or NULL
outParameters	Type: .System.Object.. The output parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

InvokeRpcMethod Overload [▶ 2107]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.InvokeRpcMethod Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object InvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifiying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifiying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.

Return Value

Type: Object
The return value of the Method (as object).

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
```

```
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

InvokeRpcMethod Overload [▶ 2107]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.2.2 IRpcCallableInstance.InvokeRpcMethodAsync Method

Overload List

	Name	Description
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]	Invokes the specified RPC Method asynchronously
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]	Invokes the specified RPC Method asynchronously

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.InvokeRpcMethodAsync Method (String, .Object., CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    CancellationToken cancel
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethodAccess [▶ 2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess [▶ 2571] results contains the return value (ReturnValue [▶ 2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode [▶ 2559]) after the communication.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
```

```
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

InvokeRpcMethodAsync Overload [▶ 2112]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.InvokeRpcMethodAsync Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken)

Invokes the specified RPC Method asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultRpcMethodAccess> InvokeRpcMethodAsync(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    CancellationToken cancel
)
```

Parameters

methodName	Type: System.String The method name.

inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
cancel	Type: System.Threading.CancellationToken The cancellation token

Return Value

Type: Task.ResultRpcMethodAccess [▶ 2571].
A task that represents the asynchronous 'InvokeRpcMethod' operation. The ResultRpcMethodAccess [▶ 2571] results contains the return value (ReturnValue [▶ 2574]) together with the output parameters. The succeeded communication is indicated by the ErrorCode property (ErrorCode [▶ 2559]) after the communication.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});
```

```
        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

InvokeRpcMethodAsync Overload [▶ 2112]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.71.2.3 IRpcCallableInstance.TryInvokeRpcMethod Method

Overload List

	Name	Description
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]	Invokes the specified RPC Method
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].

Reference

IRpcCallableInstance Interface [▶ 2104]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.TryInvokeRpcMethod Method (String, .Object., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: Int32
The result value of the call (ErrorCode). 0 means Succeeded.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */
```

```
        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

TryInvokeRpcMethod Overload [▶ 2116]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.TryInvokeRpcMethod Method (String, .Object., .Object.., Object.)

Invokes the specified RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String The method name.
inParameters	Type: .System.Object. The parameters.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method as object.

Return Value

Type: Int32
The result value of the call (ErrorCode). 0 means Succeeded.

Remarks

Because this overload doesn't provide any AnyTypeSpecifier [▶ 1633] specifications, only primitive datatypes will be correctly marshalled by this method. Complex types will fall back to byte[] arrays.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

TryInvokeRpcMethod Overload [▶ 2116]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcCallableInstance.TryInvokeRpcMethod Method (String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.)

Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryInvokeRpcMethod(
    string methodName,
    Object[] inParameters,
    AnyTypeSpecifier[] outSpecifiers,
    AnyTypeSpecifier retSpecifier,
    out Object[] outParameters,
    out Object retValue
)
```

Parameters

methodName	Type: System.String Name of the method.
inParameters	Type: .System.Object. The parameters.
outSpecifiers	Type: .TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633]. The out specifiers (specifying the out types) or NULL.
retSpecifier	Type: TwinCAT.TypeSystem.AnyTypeSpecifier [▶ 1633] The ret specifier (specifying the return value) or NULL.
outParameters	Type: .System.Object.. The out parameters.
retValue	Type: System.Object. The return value of the RPC method./>

Return Value

Type: Int32
AdsErrorCode.

Remarks

The RpcMethod optionally support In-Parameters, Out-Parameters and Return values. Therefore the parameters inParameters, outParameters, outSpecifiers, retSpecifier are allowed to be empty or NULL. In case of using primitive datatypes, the type specifier parameters (outSpecifiers and retSpecifier) are not necessary and should not be set.

Examples

The following sample shows how to call (Remote Procedures / Methods) within the PLC.

Dynamic Tree Mode

```
class RpcCallVirtualProgram
{
    /// <summary>
    /// Defines the entry point of the application.
    /// </summary>
    /// <param name="args">The arguments.</param>
    static void Main(string[] args)
    {
```

```
    // Get the AdsAddress from command-line arguments
    AmsAddress address = ArgParser.Parse(args);

    using (AdsClient client = new AdsClient())
    {
        //client.Synchronize = false;

        // Connect to the target device
        client.Connect(address);

        SymbolLoaderSettings settings = new SymbolLoaderSettings(SymbolsLoadMode.VirtualTree);
        ISymbolLoader loader = SymbolLoaderFactory.Create(client, settings);

        // Get the Symbols (Dynamic Symbols)

        IRpcStructInstance main = (IRpcStructInstance)loader.Symbols["MAIN"]; // Gets the MAIN Insta
nce of the PLC Program

        // Call a Method that has the following signature (within MAIN Program)
        /*  {attribute 'TcRpcEnable'}
        METHOD PUBLIC M_Add : INT
        VAR_INPUT
            i1 : INT := 0;
            i2 : INT := 0;
        END_VAR
        */

        short result = (short)main.InvokeRpcMethod("M_Add", new object[] {(short) 3, (short) 4});

        // Call a Method that has no parameter and returns VOID
        main.InvokeRpcMethod("M_Method1", new object[] {});

        //Browsing RpcMethods
        foreach(IRpcMethod method in main.RpcMethods)
        {
        string methodName = method.Name;

        foreach(IRpcMethodParameter parameter in method.Parameters)
        {
            string parameterName = parameter.Name;
            string parameterType = parameter.TypeName;
        }
        }
    }
    }
}
```

Reference

IRpcCallableInstance Interface [▶ 2104]

TryInvokeRpcMethod Overload [▶ 2116]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.72 IRpcCallableType Interface

Interface representing an RPC callable IStructType [▶ 2162]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcCallableType
```

The IRpcCallableType type exposes the following members.

Properties

	Name	Description
	RpcMethods [▶ 2122]	Gets the Method descriptions for the IRpcCallableType

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.72.1 IRpcCallableType Properties

The IRpcCallableType [▶ 2121] type exposes the following members.

Properties

	Name	Description
	RpcMethods [▶ 2122]	Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Reference

IRpcCallableType Interface [▶ 2121]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.72.1.1 IRpcCallableType.RpcMethods Property

Gets the Method descriptions for the IRpcCallableType [▶ 2121]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodCollection RpcMethods { get; }
```

Property Value

Type: IRpcMethodCollection [▶ 2127]
The RPC methods.

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

IRpcCallableType Interface [▶ 2121]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73 IRpcMethod Interface

Interface describes an RPC Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcMethod
```

The IRpcMethod type exposes the following members.

Properties

	Name	Description
	Comment [▶ 2124]	Gets the Method comment.
	InParameters [▶ 2124]	Gets the In-Parameters of the IRpcMethod
	IsVoid [▶ 2125]	Gets a value indicating whether this IRpcMethod has no return parameter
	Name [▶ 2125]	Gets the name of the method
	OutParameters [▶ 2126]	Gets the Out-Parameters of the IRpcMethod
	Parameters [▶ 2126]	Gets all parameters (In, Out and ref parameters) of the .
	ReturnType [▶ 2126]	Gets the return type.
	ReturnTypeSize [▶ 2127]	Gets the size of the return type in bytes.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1 IRpcMethod Properties

The IRpcMethod [▶ 2123] type exposes the following members.

Properties

	Name	Description
	Comment [▶ 2124]	Gets the Method comment.
	InParameters [▶ 2124]	Gets the In-Parameters of the IRpcMethod [▶ 2123]
	IsVoid [▶ 2125]	Gets a value indicating whether this IRpcMethod [▶ 2123] has no return parameter
	Name [▶ 2125]	Gets the name of the method

	Name	Description
	OutParameters [▶ 2126]	Gets the Out-Parameters of the IRpcMethod [▶ 2123]
	Parameters [▶ 2126]	Gets all parameters (In, Out and ref parameters) of the . [▶ 2123]
	ReturnType [▶ 2126]	Gets the return type.
	ReturnTypeSize [▶ 2127]	Gets the size of the return type in bytes.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.1 IRpcMethod.Comment Property

Gets the Method comment.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Comment { get; }
```

Property Value

Type: String
The comment.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.2 IRpcMethod.InParameters Property

Gets the In-Parameters of the IRpcMethod [▶ 2123]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodParameterCollection InParameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The In- and Ref-Parameters

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.3 IRpcMethod.IsVoid Property

Gets a value indicating whether this IRpcMethod [▶ 2123] has no return parameter

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsVoid { get; }
```

Property Value

Type: Boolean
true if this instance is void; otherwise, false.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.4 IRpcMethod.Name Property

Gets the name of the method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.5 IRpcMethod.OutParameters Property

Gets the Out-Parameters of the IRpcMethod [▶ 2123]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodParameterCollection OutParameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The out- and ref-parameters.

Remarks

The Out-Parameters doesn't include the Ref Parameters. These are included in the InParameters [▶ 2124] set.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.6 IRpcMethod.Parameters Property

Gets all parameters (In, Out and ref parameters) of the . [▶ 2123]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodParameterCollection Parameters { get; }
```

Property Value

Type: IRpcMethodParameterCollection [▶ 2137]
The parameters.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.7 IRpcMethod.ReturnType Property

Gets the return type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string ReturnType { get; }
```

Property Value

Type: String
Return type.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.73.1.8 IRpcMethod.ReturnTypeSize Property

Gets the size of the return type in bytes.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int ReturnTypeSize { get; }
```

Property Value

Type: Int32
The size of the return type.

Reference

IRpcMethod Interface [▶ 2123]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74 IRpcMethodCollection Interface

Interface for RPC Method collections.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcMethodCollection : IList<IRpcMethod>,
    ICollection<IRpcMethod>, IEnumerable<IRpcMethod>, IEnumerable
```

The IRpcMethodCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2129]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Contains(String) [▶ 2131]	Determines whether this collection contains the specified method name.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IRpcMethod [▶ 2123]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	TryGetMethod(Int32, IRpcMethod.) [▶ 2132]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2133]	Tries to get the specified method.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74.1 IRpcMethodCollection Properties

The IRpcMethodCollection [▶ 2127] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2129]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Reference

IRpcMethodCollection Interface [▶ 2127]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74.1.1 IRpcMethodCollection.Item Property

Overload List

	Name	Description
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2129]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Reference

IRpcMethodCollection Interface [▶ 2127]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodCollection.Item Property (String)

Gets the IRpcMethod [▶ 2123] with the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethod this[
    string methodName
] { get; }
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: IRpcMethod [▶ 2123]
RpcMethod.

Exceptions

Exception	Condition
KeyNotFoundException	

Reference

IRpcMethodCollection Interface [▶ 2127]

Item Overload [▶ 2129]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74.2 IRpcMethodCollection Methods

The IRpcMethodCollection [▶ 2127] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	Contains(String) [▶ 2131]	Determines whether this collection contains the specified method name.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IRpcMethod [▶ 2123]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IRpcMethod [▶ 2123]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IRpcMethod [▶ 2123]..)

	Name	Description
	TryGetMethod(Int32, IRpcMethod.) [▶ 2132]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2133]	Tries to get the specified method.

Reference

IRpcMethodCollection Interface [▶ 2127]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74.2.1 IRpcMethodCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2131]	Determines whether this collection contains the specified method name.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethod [▶ 2123]..)

Reference

IRpcMethodCollection Interface [▶ 2127]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodCollection.Contains Method (String)

Determines whether this collection contains the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string methodName
)
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: Boolean
true if contained.; otherwise, false.

Reference

IRpcMethodCollection Interface [▶ 2127]

Contains Overload [▶ 2131]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.74.2.2 IRpcMethodCollection.TryGetMethod Method

Overload List

	Name	Description
	TryGetMethod(Int32, IRpcMethod.) [▶ 2132]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2133]	Tries to get the specified method.

Reference

IRpcMethodCollection Interface [▶ 2127]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

Parameters

vTableIndex	Type: System.Int32 vTableIndex.
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Reference

IRpcMethodCollection Interface [▶ 2127]

TryGetMethod Overload [▶ 2132]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetMethod(
    string methodName,
    out IRpcMethod method
)
```

Parameters

methodName	Type: System.String Name of the method.
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Reference

IRpcMethodCollection Interface [▶ 2127]

TryGetMethod Overload [▶ 2132]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75 IRpcMethodParameter Interface

Interface IRpcMethodParameter

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcMethodParameter
```

The IRpcMethodParameter type exposes the following members.

Properties

	Name	Description
	HasLengthIsParameter [▶ 2135]	Gets a value indicating whether this instance has a related LengthIs Parameter.
	LengthIsParameterIndex [▶ 2135]	Gets the index of the LengthIs parameter (within the MethodParameter List)
	Name [▶ 2135]	Gets the Parameter Name
	ParameterFlags [▶ 2136]	Gets the parameter flags.
	Size [▶ 2136]	Gets the size of the IRpcMethodParameter
	TypeName [▶ 2137]	Gets the Data type of the Parameter

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1 IRpcMethodParameter Properties

The IRpcMethodParameter [▶ 2133] type exposes the following members.

Properties

	Name	Description
	HasLengthIsParameter [▶ 2135]	Gets a value indicating whether this instance has a related LengthIs Parameter.
	LengthIsParameterIndex [▶ 2135]	Gets the index of the LengthIs parameter (within the MethodParameter List)
	Name [▶ 2135]	Gets the Parameter Name
	ParameterFlags [▶ 2136]	Gets the parameter flags.
	Size [▶ 2136]	Gets the size of the IRpcMethodParameter [▶ 2133]
	TypeName [▶ 2137]	Gets the Data type of the Parameter

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.1 IRpcMethodParameter.HasLengthIsParameter Property

Gets a value indicating whether this instance has a related LengthIs Parameter.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool HasLengthIsParameter { get; }
```

Property Value

Type: Boolean
true if this instance has a LengthIs parameter; otherwise, false.

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.2 IRpcMethodParameter.LengthIsParameterIndex Property

Gets the index of the LengthIs parameter (within the MethodParameter List)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int LengthIsParameterIndex { get; }
```

Property Value

Type: Int32
The index of the length is parameter.

Remarks

This field references to the Parameter that defines the length for this generic one. Equally to the marshalling attributes of COM (sizeof, length) this enables to transport parameter of type (PVOID)

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.3 IRpcMethodParameter.Name Property

Gets the Parameter Name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.4 IRpcMethodParameter.ParameterFlags Property

Gets the parameter flags.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
MethodParamFlags ParameterFlags { get; }
```

Property Value

Type: MethodParamFlags [▶ 2288]
The parameter flags.

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.5 IRpcMethodParameter.Size Property

Gets the size of the IRpcMethodParameter [▶ 2133]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Size { get; }
```

Property Value

Type: Int32
The size.

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.75.1.6 IRpcMethodParameter.TypeName Property

Gets the Data type of the Parameter

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string TypeName { get; }
```

Property Value

Type: String
The type.

Reference

IRpcMethodParameter Interface [▶ 2133]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.76 IRpcMethodParameterCollection Interface

Interface IRpcMethodParameterCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcMethodParameterCollection : IList<IRpcMethodParameter>,
    ICollection<IRpcMethodParameter>, IEnumerable<IRpcMethodParameter>, IEnumerable
```

The IRpcMethodParameterCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)

	Name	Description
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Item	Gets or sets the element at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IRpcMethodParameter [▶ 2133]..)
	GetLengthIsParameter [▶ 2139]	Gets the corresponding LengthIs parameter.
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.76.1 IRpcMethodParameterCollection Properties

The IRpcMethodParameterCollection [▶ 2137] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Item	Gets or sets the element at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)

Reference

IRpcMethodParameterCollection Interface [▶ 2137]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.76.2 IRpcMethodParameterCollection Methods

The IRpcMethodParameterCollection [▶ 2137] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	Contains	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.IRpcMethodParameter [▶ 2133]..)
	GetLengthIsParameter [▶ 2139]	Gets the corresponding LengthIs parameter.
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.IRpcMethodParameter [▶ 2133]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.IRpcMethodParameter [▶ 2133]..)

Reference

IRpcMethodParameterCollection Interface [▶ 2137]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.76.2.1 IRpcMethodParameterCollection.GetLengthIsParameter Method

Gets the corresponding LengthIs parameter.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```

Parameters

parameter	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The value parameter

Return Value

Type: IRpcMethodParameter [▶ 2133]
The LengthIs Parameter

Reference

IRpcMethodParameterCollection Interface [▶ 2137]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodParameter.LengthIsParameterIndex [▶ 2135]

IRpcMethodParameter.HasLengthIsParameter [▶ 2135]

6.11.77 IRpcStructInstance Interface

Interface IRpcStructInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IRpcStructInstance : IStructInstance,
    ISymbol, IAttributedInstance, IInstance, IBitSize, IRpcCallableInstance
```

The IRpcStructInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [▶ 2158].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)

	Name	Description
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from IStructInstance [▶ 2158].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	RpcMethods [▶ 2105]	Gets the Method descriptions for the IRpcCallableType [▶ 2121] (Inherited from IRpcCallableInstance [▶ 2104].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 2107]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)

	Name	Description
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]	Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]	Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [▶ 2104].)
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104]. (Inherited from IRpcCallableInstance [▶ 2104].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IStructInstance [▶ 2158]

TwinCAT.TypeSystem.IRpcCallableInstance [▶ 2104]

6.11.77.1 IRpcStructInstance Properties

The IRpcStructInstance [▶ 2140] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [▶ 2158].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from IStructInstance [▶ 2158].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	RpcMethods [▶ 2105]	Gets the Method descriptions for the IRpcCallableType [▶ 2121] (Inherited from IRpcCallableInstance [▶ 2104].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)

	Name	Description
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IRpcStructInstance Interface [▶ 2140]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.77.2 IRpcStructInstance Methods

Methods

	Name	Description
	InvokeRpcMethod(String, .Object.) [▶ 2107]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethod(String, .Object., .Object..) [▶ 2109]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object..) [▶ 2110]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethodAsync(String, .Object., CancellationToken) [▶ 2113]	Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [▶ 2104].)
	InvokeRpcMethodAsync(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, CancellationToken) [▶ 2114]	Invokes the specified RPC Method asynchronously (Inherited from IRpcCallableInstance [▶ 2104].)
	TryInvokeRpcMethod(String, .Object., Object.) [▶ 2116]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)
	TryInvokeRpcMethod(String, .Object., .Object.., Object.) [▶ 2118]	Invokes the specified RPC Method (Inherited from IRpcCallableInstance [▶ 2104].)

	Name	Description
	TryInvokeRpcMethod(String, .Object., .AnyTypeSpecifier., AnyTypeSpecifier, .Object.., Object.) [▶ 2120]	Invokes the the specified RpcMethod of the IRpcCallableInstance [▶ 2104]. (Inherited from IRpcCallableInstance [▶ 2104].)

Reference

IRpcStructInstance Interface [▶ 2140]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.78 IStringInstance Interface

Interface IStringInstance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStringInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IStringInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsFixedLength [▶ 2148]	Gets a value indicating whether this instance is a string of static length
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

TwinCAT.TypeSystem.IStringType [▶ 2155]

6.11.78.1 IStringInstance Properties

The IStringInstance [▶ 2145] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)

	Name	Description
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsFixedLength [▶ 2148]	Gets a value indicating whether this instance is a string of static length
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IStringInstance Interface [▶ 2145]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.78.1.1 IStringInstance.IsFixedLength Property

Gets a value indicating whether this instance is a string of static length

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsFixedLength { get; }
```

Property Value

Type: Boolean
true if this instance is of static length; otherwise, false.

Reference

IStringInstance Interface [▶ 2145]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.79 IStringMarshaler Interface

Common interface for marshalling ADS string values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStringMarshaler
```

Methods

	Name	Description
	Marshal(String, Span) [▶ 2150]	
	Marshal(IStringType, String, Span) [▶ 2150]	
	MarshalSize(String) [▶ 2151]	Gets the marshal size of the string.
	MarshalSize(IStringType) [▶ 2152]	Gets the marshal size of the specified string type.

	Name	Description
	MarshalSize(Encoding, Int32) [▶ 2153]	Gets the marshal size of the string given by its length.
	Unmarshal(ReadOnlySpan, Void, Byte) [▶ 2154]	
	Unmarshal(IStringType, ReadOnlySpan, Void) [▶ 2154]	

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.79.1 IStringMarshaler Methods

Methods

	Name	Description
	Marshal(String, Span) [▶ 2150]	
	Marshal(IStringType, String, Span) [▶ 2150]	
	MarshalSize(String) [▶ 2151]	Gets the marshal size of the string.
	MarshalSize(IStringType) [▶ 2152]	Gets the marshal size of the specified string type.
	MarshalSize(Encoding, Int32) [▶ 2153]	Gets the marshal size of the string given by its length.
	Unmarshal(ReadOnlySpan, Void, Byte) [▶ 2154]	
	Unmarshal(IStringType, ReadOnlySpan, Void) [▶ 2154]	

Reference

IStringMarshaler Interface [▶ 2148]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.79.1.1 IStringMarshaler.Marshal Method

Overload List

	Name	Description
	Marshal(String, Span) [▶ 2150]	
	Marshal(IStringType, String, Span) [▶ 2150]	

Reference

IStringMarshaler Interface [▶ 2148]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.Marshal Method (String, Span`1)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Marshal(
    string value,
    Span destination
)
```

Parameters

value	Type: System.String
destination	Type: Span

Return Value

Type: Int32

Reference

IStringMarshaler Interface [▶ 2148]

Marshal Overload [▶ 2150]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.Marshal Method (IStringType, String, Span`1)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Marshal(
    IStringType type,
    string value,
    Span destination
)
```

Parameters

type	Type: TwinCAT.TypeSystem.IStringType [▶ 2155]
value	Type: System.String
destination	Type: Span

Return Value

Type: Int32

Reference

IStringMarshaler Interface [▶ 2148]

Marshal Overload [▶ 2150]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.79.1.2 IStringMarshaler.MarshalSize Method

Overload List

	Name	Description
	MarshalSize(String) [▶ 2151]	Gets the marshal size of the string.
	MarshalSize(IStringType) [▶ 2152]	Gets the marshal size of the specified string type.
	MarshalSize(Encoding, Int32) [▶ 2153]	Gets the marshal size of the string given by its length.

Reference

IStringMarshaler Interface [▶ 2148]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.MarshalSize Method (String)

Gets the marshal size of the string.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int MarshalSize(
    string value
)
```

Parameters

value	Type: System.String The string value.

Return Value

Type: Int32
Marshalling size of the string.

Reference

IStringMarshaler Interface [▶ 2148]

MarshalSize Overload [▶ 2151]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.MarshalSize Method (IStringType)

Gets the marshal size of the specified string type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int MarshalSize(
    IStringType stringType
)
```

Parameters

stringType	Type: TwinCAT.TypeSystem.IStringType [▶ 2155] Type of the string.

Return Value

Type: Int32
Marshalling size of the string

Reference

IStringMarshaler Interface [▶ 2148]

MarshalSize Overload [▶ 2151]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.MarshalSize Method (Encoding, Int32)

Gets the marshal size of the string given by its length.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int MarshalSize(
    Encoding encoding,
    int strLen
)
```

Parameters

encoding	Type: System.Text.Encoding The encoding.
strLen	Type: System.Int32 Length of the string.

Return Value

Type: Int32
Marshalling size of the string.

Reference

IStringMarshaler Interface [▶ 2148]

MarshalSize Overload [▶ 2151]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.79.1.3 IStringMarshaler.Unmarshal Method

Overload List

	Name	Description
	Unmarshal(ReadOnlySpan, Void, Byte) [▶ 2154]	
	Unmarshal(IStringType, ReadOnlySpan, Void) [▶ 2154]	

Reference

IStringMarshaler Interface [▶ 2148]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.Unmarshal Method (ReadOnlySpan`1, Void, Byte)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Unmarshal(
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters

source	Type: ReadOnlySpan
encoding	Type: System.Void
value	Type: System.Byte

Return Value

Type: Int32

Reference

IStringMarshaler Interface [▶ 2148]

Unmarshal Overload [▶ 2153]

TwinCAT.TypeSystem Namespace [▶ 1622]

IStringMarshaler.Unmarshal Method (IStringType, ReadOnlySpan`1, Void)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Unmarshal(
    IStringType stringType,
    ReadOnlySpan source,
    void value
)
```

Parameters

stringType	Type: TwinCAT.TypeSystem.IStringType [▶ 2155]
source	Type: ReadOnlySpan

value Type: System.Void

Return Value

Type: Int32

Reference

IStringMarshaler Interface [▶ 2148]

Unmarshal Overload [▶ 2153]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.80 IStringType Interface

Interface representing a string IDataType [▶ 1986]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStringType : IDataType,
    IBitSize
```

The IStringType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Encoding [▶ 2157]	Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsFixedLength [▶ 2157]	Gets a value indicating whether the string is of fixed length.
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Length [▶ 2158]	Gets the number of characters within the string (when fixed length).
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.80.1 IStringType Properties

The IStringType [▶ 2155] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Encoding [▶ 2157]	Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsFixedLength [▶ 2157]	Gets a value indicating whether the string is of fixed length.
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Length [▶ 2158]	Gets the number of characters within the string (when fixed length).
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IStringType Interface [▶ 2155]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.80.1.1 IStringType.Encoding Property

Gets the encoding of the String (Encoding.Default (STRING) or Encoding.UNICODE (WSTRING))

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Encoding Encoding { get; }
```

Property Value

Type: Encoding
The encoding.

Reference

IStringType Interface [▶ 2155]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.80.1.2 IStringType.IsFixedLength Property

Gets a value indicating whether the string is of fixed length.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsFixedLength { get; }
```

Property Value

Type: Boolean
true if this instance is fixed length; otherwise, false.

Reference

IStringType Interface [▶ 2155]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.80.1.3 IStringType.Length Property

Gets the number of characters within the string (when fixed length).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Length { get; }
```

Property Value

Type: Int32
The length if fixed length, otherwise -1

Reference

IStringType Interface [▶ 2155]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.81 IStructInstance Interface

Interface representing an instance of a IStructType [▶ 2162]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStructInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IStructInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

	Name	Description
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.81.1 IStructInstance Properties

The IStructInstance [▶ 2158] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance [▶ 2158].
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IStructInstance Interface [▶ 2158]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.81.1.1 IStructInstance.HasRpcMethods Property

Gets a value indicating whether this instance has RPC methods

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this instance has RPC methods; otherwise, false.

Remarks

If the struct instance supports RPC Methods, then the instance class is also supporting IRpcStructInstance [▶ 2140]:

Reference

IStructInstance Interface [▶ 2158]

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IRpcStructInstance [▶ 2140]

TwinCAT.TypeSystem.IRpcMethod [▶ 2123]

TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133]

6.11.81.1.2 IStructInstance.MemberInstances Property

Gets the member instances of the Struct Instance [▶ 2158].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> MemberInstances { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The member instances.

Reference

IStructInstance Interface [▶ 2158]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82 IStructType Interface

Interface representing Struct data types

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStructType : IDataType,
    IBitSize
```

The IStructType type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 2165]	Gets all members (down the derivation hierarchy)
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 2165]	Gets the structs Base Type (Null if not derived).

	Name	Description
	BaseTypeName [▶ 2165]	Gets the the Name of the Base class (if derived)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	HasRpcMethods [▶ 2166]	Gets a value indicating whether this instance has RPC methods (Struct types only)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Members [▶ 2166]	Gets a readonly collection of the Members [▶ 2065] of the IStructType.
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1 IStructType Properties

The IStructType [▶ 2162] type exposes the following members.

Properties

	Name	Description
	AllMembers [▶ 2165]	Gets all members (down the derivation hierarchy)

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 2165]	Gets the structs Base Type (Null if not derived).
	BaseTypeName [▶ 2165]	Gets the the Name of the Base class (if derived)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	HasRpcMethods [▶ 2166]	Gets a value indicating whether this instance has RPC methods (Struct types only)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Members [▶ 2166]	Gets a readonly collection of the Members [▶ 2065] of the IStructType [▶ 2162].
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1.1 IStructType.AllMembers Property

Gets all members (down the derivation hierarchy)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IMemberCollection AllMembers { get; }
```

Property Value

Type: IMemberCollection [▶ 2068]
All members.

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1.2 IStructType.BaseType Property

Gets the structs Base Type (Null if not derived).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1.3 IStructType.BaseTypeName Property

Gets the the Name of the Base class (if derived)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string BaseTypeName { get; }
```

Property Value

Type: String
Empty if not derived.

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1.4 IStructType.HasRpcMethods Property

Gets a value indicating whether this instance has RPC methods (Struct types only)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool HasRpcMethods { get; }
```

Property Value

Type: Boolean
true if this instance has RPC methods; otherwise, false.

Remarks

The DataType (Structure) must be marked with the PlcAttribute 'TcRpcEnable' to enable RpcMethods, otherwise RpcMethods are not passed through to the ADS symbolic information.

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.82.1.5 IStructType.Members Property

Gets a readonly collection of the Members [▶ 2065] of the IStructType [▶ 2162].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IMemberCollection Members { get; }
```

Property Value

Type: IMemberCollection [▶ 2068]
The members as readonly collection.

Remarks

If the IStructType [▶ 2162] is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers [▶ 2165] property.

Reference

IStructType Interface [▶ 2162]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.83 IStructValue Interface

Interface IStructValue

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IStructValue : IValue
```

The IStructValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)

	Name	Description
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TryGetMemberValue [▶ 2169]	Tries to get a property/Member value.
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TrySetMemberValue [▶ 2170]	Tries to Set a Member/Property Value
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IValue [▶ 2226]

6.11.83.1 IStructValue Properties

The IStructValue [▶ 2167] type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value) (Inherited from IValue [▶ 2226].)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data. (Inherited from IValue [▶ 2226].)
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226] (Inherited from IValue [▶ 2226].)
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value. (Inherited from IValue [▶ 2226].)
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226]. (Inherited from IValue [▶ 2226].)
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC) (Inherited from IValue [▶ 2226].)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet) (Inherited from IValue [▶ 2226].)

Reference

IStructValue Interface [▶ 2167]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.83.2 IStructValue Methods

The IStructValue [▶ 2167] type exposes the following members.

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ReadAsync [▶ 2231]	Reads the value (via ADS) (Inherited from IValue [▶ 2226].)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TryGetMemberValue [▶ 2169]	Tries to get a property/Member value.
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value. (Inherited from IValue [▶ 2226].)
	TrySetMemberValue [▶ 2170]	Tries to Set a Member/Property Value
	Write [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)
	WriteAsync [▶ 2233]	Writes the value (via ADS) (Inherited from IValue [▶ 2226].)

Reference

IStructValue Interface [▶ 2167]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.83.2.1 IStructValue.TryGetMemberValue Method

Tries to get a property/Member value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetMemberValue(
    string name,
    out Object value
)
```

Parameters

name	Type: System.String The name of the member
value	Type: System.Object. The value.

Return Value

Type: Boolean
true if succeeded, otherwise false otherwise.

Reference

IStructValue Interface [▶ 2167]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.83.2.2 IStructValue.TrySetMemberValue Method

Tries to Set a Member/Property Value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TrySetMemberValue(
    string name,
    Object value
)
```

Parameters

name	Type: System.String The name of the member
value	Type: System.Object The value.

Return Value

Type: Boolean
true if succeeded, otherwise false otherwise.

Reference

IStructValue Interface [▶ 2167]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.84 ISubRangeType Interface

Interface representing a SubRange type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISubRangeType : IDataType,
    IBitSize
```

The ISubRangeType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)

	Name	Description
	BaseType [▶ 2172]	Gets the the base type of the ISubRangeType
	BaseTypeName [▶ 2173]	Gets the name of the base type.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.84.1 ISubRangeType Properties

The ISubRangeType [▶ 2170] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 2172]	Gets the the base type of the ISubRangeType [▶ 2170]

	Name	Description
	BaseTypeName [▶ 2173]	Gets the name of the base type.
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

ISubRangeType Interface [▶ 2170]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.84.1.1 ISubRangeType.BaseType Property

Gets the the base type of the ISubRangeType [▶ 2170]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType BaseType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The base datat type or **null** if not resolved.

Reference

ISubRangeType Interface [▶ 2170]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.84.1.2 ISubRangeType.BaseTypeName Property

Gets the name of the base type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string BaseTypeName { get; }
```

Property Value

Type: String
The name of the base type.

Reference

ISubRangeType Interface [▶ 2170]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.85 ISubRangeType.T. Interface

Interface representing a SubRange type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISubRangeType<T> : ISubRangeType,
    IDataType, IBitSize
where T : struct, new()
```

Type Parameters

T

The ISubRangeType.T. type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 2172]	Gets the the base type of the ISubRangeType [▶ 2170] (Inherited from ISubRangeType [▶ 2170].)
	BaseTypeName [▶ 2173]	Gets the name of the base type. (Inherited from ISubRangeType [▶ 2170].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	LowerBound [▶ 2176]	Gets the lower bound.
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	UpperBound [▶ 2176]	Gets the upper bound.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.85.1 ISubRangeType.T. Properties

The ISubRangeType.T. [▶ 2173] generic type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BaseType [▶ 2172]	Gets the the base type of the ISubRangeType [▶ 2170] (Inherited from ISubRangeType [▶ 2170].)
	BaseTypeName [▶ 2173]	Gets the name of the base type. (Inherited from ISubRangeType [▶ 2170].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	LowerBound [▶ 2176]	Gets the lower bound.
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	UpperBound [▶ 2176]	Gets the upper bound.

Reference

ISubRangeType.T. Interface [▶ 2173]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.85.1.1 ISubRangeType.T..LowerBound Property

Gets the lower bound.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T LowerBound { get; }
```

Property Value

Type: T [▶ 2173]
The lower bound.

Reference

ISubRangeType.T. Interface [▶ 2173]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.85.1.2 ISubRangeType.T..UpperBound Property

Gets the upper bound.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
T UpperBound { get; }
```

Property Value

Type: T [▶ 2173]
The upper bound.

Reference

ISubRangeType.T. Interface [▶ 2173]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86 ISymbol Interface

Interface specifying Symbols (

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbol : IAttributedInstance,
    IInstance, IBitSize
```

The ISymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type.
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol is persistent.
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type.
	IsReadOnly [▶ 2181]	Indicates that this instance is read only.
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive.
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol

	Name	Description
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1 ISymbol Properties

The ISymbol [▶ 2176] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type.
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent.
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type.
	IsReadOnly [▶ 2181]	Indicates that this instance is read only.

	Name	Description
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive.
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176]
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.1 ISymbol.Category Property

Gets the Symbol/Datatype Category

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DataTypeCategory Category { get; }
```

Property Value

Type: DataTypeCategory [▶ 1649]
The category.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.2 ISymbol.IsContainerType Property

Gets a value indicating whether this Symbol is acontainer type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsContainerType { get; }
```

Property Value

Type: Boolean
true if this instance is container type; otherwise, false.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.3 ISymbol.IsPersistent Property

Gets a value indicating whether this ISymbol [▶ 2176] is persistent.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPersistent { get; }
```

Property Value

Type: Boolean
true if this instance is persistent; otherwise, false.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.4 ISymbol.IsPrimitiveType Property

Gets a value indicating whether this instance is a primitive type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPrimitiveType { get; }
```

Property Value

Type: Boolean
true if this instance is primitive type; otherwise, false.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.5 ISymbol.IsReadOnly Property

Indicates that this instance is read only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsReadOnly { get; }
```

Property Value

Type: Boolean

Remarks

Actually, this Flag is restricted to TcCOM-Objects readonly Parameters. Within the PLC this is used for the ApplicationName and ProjectName of PLC instances. Write-Access on these Modules will create an DeviceAccessDenied [▶ 575] error.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.6 ISymbol.IsRecursive Property

Gets a value indicating whether this instance is recursive.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsRecursive { get; }
```

Property Value

Type: Boolean
true if this instance is recursive; otherwise, false.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.7 ISymbol.Parent Property

Gets the parent Symbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol Parent { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The parent.

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.86.1.8 ISymbol.SubSymbols Property

Gets the SubSymbols of the ISymbol [▶ 2176]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> SubSymbols { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].

Remarks

Used for Array, Struct, Pointer and Reference instances. Otherwise empty

Reference

ISymbol Interface [▶ 2176]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.87 ISymbolCollection Interface

Interface ISymbolCollection Implements the IInstanceCollection.T. [▶ 2057]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolCollection : ISymbolCollection<ISymbol>,
    IInstanceCollection<ISymbol>, IList<ISymbol>, ICollection<ISymbol>,
    IEnumerable<ISymbol>, IEnumerable
```

The ISymbolCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.ISymbol [▶ 2176]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.ISymbol [▶ 2176]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.ISymbol [▶ 2176]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ISymbol [▶ 2176]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.ISymbol [▶ 2176]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)

	Name	Description
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IInstanceCollection.T. [▶ 2057]

6.11.87.1 ISymbolCollection Properties

The ISymbolCollection [▶ 2182] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.ISymbol [▶ 2176]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

ISymbolCollection Interface [▶ 2182]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.87.2 ISymbolCollection Methods

The ISymbolCollection [▶ 2182] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)

	Name	Description
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.ISymbol [▶ 2176]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.ISymbol [▶ 2176]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ISymbol [▶ 2176]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.ISymbol [▶ 2176]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.ISymbol [▶ 2176]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.ISymbol [▶ 2176]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

ISymbolCollection Interface [▶ 2182]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.88 ISymbolCollection.T. Interface

Interface ISymbolCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolCollection<T> : IInstanceCollection<T>,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, ISymbol
```

Type Parameters

T

The ISymbolCollection.T. type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.T..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.T..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.T..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.T..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IInstanceCollection.T. [▶ 2057]

6.11.88.1 ISymbolCollection.T. Properties

The ISymbolCollection.T. [▶ 2185] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.T [▶ 2185]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.T [▶ 2185]..)
	Item.String. [▶ 2059]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.T [▶ 2185]..)
	Mode [▶ 2060]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

ISymbolCollection.T. Interface [▶ 2185]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.88.2 ISymbolCollection.T. Methods

The ISymbolCollection.T. [▶ 2185] generic type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.T [▶ 2185]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.T [▶ 2185]..)
	Contains(String) [▶ 2061]	Determines whether this collection contains an instance with the specified instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.T [▶ 2185]..)
	ContainsName [▶ 2062]	Determines whether this collection contains an instance with the specified instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.T [▶ 2185]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.T [▶ 2185]..)
	GetInstance [▶ 2062]	Gets the IInstance [▶ 2052]by instance path. (Inherited from IInstanceCollection.T. [▶ 2057].)
	GetInstanceByName [▶ 2063]	Gets the IInstance [▶ 2052] by instance name. (Inherited from IInstanceCollection.T. [▶ 2057].)

	Name	Description
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.T [▶ 2185]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.T [▶ 2185]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.T [▶ 2185]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.T [▶ 2185]..)
	TryGetInstance [▶ 2063]	Tries to get the specified instance. (Inherited from IInstanceCollection.T. [▶ 2057].)
	TryGetInstanceByName [▶ 2064]	Tries to get the specified instance by name. (Inherited from IInstanceCollection.T. [▶ 2057].)

Reference

ISymbolCollection.T. Interface [▶ 2185]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89 ISymbolFactory Interface

Symbol Factory Interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolFactory
```

The ISymbolFactory type exposes the following members.

Properties

	Name	Description
	FactoryServices [▶ 2189]	Gets the factory services.
	HasInvalidCharacters [▶ 2190]	Gets a value indicating whether ISymbol [▶ 2176]s have invalid characters
	InvalidCharacters [▶ 2190]	Gets the invalid characters that are not allowed to appear within the Instance Name

Methods

	Name	Description
	CreateArrayElement [▶ 2192]	Creates a single Array Element

	Name	Description
	CreateArrayElementInstances [▶ 2192]	Creates all Element Instances of the specified array parent symbol.
	CreateFieldInstance [▶ 2193]	Creates a singgle Instance member on a struct parent
	CreateFieldInstances [▶ 2194]	Creates the Member Instances collection for the specified parent instance
	CreateInstance [▶ 2194]	Creates the Symbol with the specified resolver
	CreateInstanceAsync [▶ 2195]	Creates the Symbol with the specified resolver
	CreateReferenceInstance [▶ 2195]	Creates the dereferenced Pointer instance
	CreateVirtualStruct [▶ 2196]	Creates the virtual structure.
	Initialize [▶ 2197]	Initializes the the ISymbolFactory.
	SetInvalidCharacters [▶ 2197]	Sets the invalid characters.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.1 ISymbolFactory Properties

The ISymbolFactory [▶ 2188] type exposes the following members.

Properties

	Name	Description
	FactoryServices [▶ 2189]	Gets the factory services.
	HasInvalidCharacters [▶ 2190]	Gets a value indicating whether ISymbol [▶ 2176]s have invalid characters
	InvalidCharacters [▶ 2190]	Gets the invalid characters that are not allowed to appear within the Instance Name

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.1.1 ISymbolFactory.FactoryServices Property

Gets the factory services.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolFactoryServices FactoryServices { get; }
```

Property Value

Type: ISymbolFactoryServices
The factory services.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.1.2 ISymbolFactory.HasInvalidCharacters Property

Gets a value indicating whether ISymbol [▶ 2176]s have invalid characters

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool HasInvalidCharacters { get; }
```

Property Value

Type: Boolean
true if this instance has invalid characters; otherwise, false.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolFactory.InvalidCharacters [▶ 2190]

6.11.89.1.3 ISymbolFactory.InvalidCharacters Property

Gets the invalid characters that are not allowed to appear within the Instance Name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
char[] InvalidCharacters { get; }
```

Property Value

Type: .Char.
The forbidden characters.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolFactory.SetInvalidCharacters(.Char.) [▶ 2197]

ISymbolFactory.HasInvalidCharacters [▶ 2190]

6.11.89.2 ISymbolFactory Methods

The ISymbolFactory [▶ 2188] type exposes the following members.

Methods

	Name	Description
	CreateArrayElement [▶ 2192]	Creates a single Array Element
	CreateArrayElementInstances [▶ 2192]	Creates all Element Instances of the specified array parent symbol.
	CreateFieldInstance [▶ 2193]	Creates a singgle Instance member on a struct parent
	CreateFieldInstances [▶ 2194]	Creates the Member Instances collection for the specified parent instance
	CreateInstance [▶ 2194]	Creates the Symbol with the specified resolver
	CreateInstanceAsync [▶ 2195]	Creates the Symbol with the specified resolver
	CreateReferenceInstance [▶ 2195]	Creates the dereferenced Pointer instance
	CreateVirtualStruct [▶ 2196]	Creates the virtual structure.
	Initialize [▶ 2197]	Initializes the the ISymbolFactory [▶ 2188].
	SetInvalidCharacters [▶ 2197]	Sets the invalid characters.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.1 ISymbolFactory.CreateArrayElement Method

Creates a single Array Element

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol CreateArrayElement(
    IArrayType arrayType,
    int[] currentIndex,
    ISymbol parent
)
```

Parameters

arrayType	Type: TwinCAT.TypeSystem.IArrayType [▶ 1971] Resolved Array type.
currentIndex	Type: .System.Int32. Array Index of the Element
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] Array Instance

Return Value

Type: ISymbol [▶ 2176]
Array element

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.2 ISymbolFactory.CreateArrayElementInstances Method

Creates all Element Instances of the specified array parent symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> CreateArrayElementInstances(
    ISymbol parentInstance,
    IArrayType arrayType
)
```

Parameters

parentInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent instance.
arrayType	Type: TwinCAT.TypeSystem.IArrayType [▶ 1971] Resolved array type.

Return Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
SymbolCollection.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.3 ISymbolFactory.CreateFieldInstance Method

Creates a singgle Instance member on a struct parent

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol CreateFieldInstance(
    IField field,
    ISymbol parent
)
```

Parameters

field	Type: TwinCAT.TypeSystem.IField [▶ 2040] Field
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] Parent Struct/Alias/Union

Return Value

Type: ISymbol [▶ 2176]
Instance member

Remarks

Because the Alias type can act like a struct, the parent can be an IAliasInstance also.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.4 ISymbolFactory.CreateFieldInstances Method

Creates the Member Instances collection for the specified parent instance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> CreateFieldInstances(
    ISymbol parentInstance,
    IDataType parentType
)
```

Parameters

parentInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent instance.
parentType	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] Parent Type (Struct/Alias/Union).

Return Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
SymbolCollection.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.5 ISymbolFactory.CreateInstance Method

Creates the Symbol with the specified resolver

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol CreateInstance(
    ISymbolInfo entry,
    ISymbol parent
)
```

Parameters

entry	Type: TwinCAT.TypeSystem.ISymbolInfo [▶ 2199] Symbol Entry.
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent.

Return Value

Type: ISymbol [▶ 2176]
Symbol instance.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.6 ISymbolFactory.CreateInstanceAsync Method

Creates the Symbol with the specified resolver

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultValue<ISymbol>> CreateInstanceAsync(
    ISymbolInfo entry,
    ISymbol parent,
    CancellationToken cancel
)
```

Parameters

entry	Type: TwinCAT.TypeSystem.ISymbolInfo [▶ 2199] Symbol Entry.
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultValue [▶ 1029].ISymbol [▶ 2176]..
Task<ResultValue<ISymbol>>.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.7 ISymbolFactory.CreateReferenceInstance Method

Creates the dereferenced Pointer instance

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol CreateReferenceInstance(
    IPointerType type,
    ISymbol parent
)
```

Parameters

type	Type: TwinCAT.TypeSystem.IPointerType [▶ 2086] Reference/Pointer type.
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] Parent Instance of the reference

Return Value

Type: ISymbol [▶ 2176]
Reference/Pointer instance.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.8 ISymbolFactory.CreateVirtualStruct Method

Creates the virtual structure.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol CreateVirtualStruct(
    string instanceName,
    string instancePath,
    ISymbol parent
)
```

Parameters

instanceName	Type: System.String Name of the instance.
instancePath	Type: System.String The instance path.
parent	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The parent.

Return Value

Type: ISymbol [▶ 2176]
Virtual struct instance

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.9 ISymbolFactory.Initialize Method

Initializes the the ISymbolFactory [▶ 2188].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Initialize(
    ISymbolFactoryServices services
)
```

Parameters

services	Type: ISymbolFactoryServices The services.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.89.2.10 ISymbolFactory.SetInvalidCharacters Method

Sets the invalid characters.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void SetInvalidCharacters(
    char[] invalidChars
)
```

Parameters

invalidChars	Type: .System.Char. The character.

Reference

ISymbolFactory Interface [▶ 2188]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolFactory.InvalidCharacters [▶ 2190]

6.11.90 ISymbolFactoryServicesProvider Interface

Symbol Value Access interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolFactoryServicesProvider
```

The ISymbolFactoryServicesProvider type exposes the following members.

Properties

	Name	Description
	FactoryServices [▶ 2198]	Gets the factory services.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.90.1 ISymbolFactoryServicesProvider Properties

The ISymbolFactoryServicesProvider [▶ 2198] type exposes the following members.

Properties

	Name	Description
	FactoryServices [▶ 2198]	Gets the factory services.

Reference

ISymbolFactoryServicesProvider Interface [▶ 2198]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.90.1.1 ISymbolFactoryServicesProvider.FactoryServices Property

Gets the factory services.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolFactoryServices FactoryServices { get; }
```

Property Value

Type: ISymbolFactoryServices
The factory services.

Reference

ISymbolFactoryServicesProvider Interface [▶ 2198]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.91 ISymbolInfo Interface

Interface ISymbolInfo

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolInfo
```

The ISymbolInfo type exposes the following members.

Properties

	Name	Description
	InstancePath [▶ 2200]	Gets the Symbol Path
	TypeName [▶ 2200]	Gets the data type Name

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.91.1 ISymbolInfo Properties

The ISymbolInfo [▶ 2199] type exposes the following members.

Properties

	Name	Description
	InstancePath [▶ 2200]	Gets the Symbol Path
	TypeName [▶ 2200]	Gets the data type Name

Reference

ISymbolInfo Interface [▶ 2199]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.91.1.1 ISymbolInfo.InstancePath Property

Gets the Symbol Path

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string InstancePath { get; }
```

Property Value

Type: String
The path.

Reference

ISymbolInfo Interface [▶ 2199]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.91.1.2 ISymbolInfo.TypeName Property

Gets the data type Name

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string TypeName { get; }
```

Property Value

Type: String
The type of the data.

Reference

ISymbolInfo Interface [▶ 2199]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.92 ISymbolLoader Interface

Symbol Loader interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolLoader : ISymbolProvider,
    ISymbolServer
```

The ISymbolLoader type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types.
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)
	Settings [▶ 2202]	Gets or sets the access Method
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.92.1 ISymbolLoader Properties

The ISymbolLoader [▶ 2200] type exposes the following members.

Properties

	Name	Description
	BuildInTypes [▶ 2202]	Gets the build in types.
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace (Inherited from ISymbolProvider [▶ 2203].)

	Name	Description
	Settings [▶ 2202]	Gets or sets the access Method
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Reference

ISymbolLoader Interface [▶ 2200]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.92.1.1 ISymbolLoader.BuildInTypes Property

Gets the build in types.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataTypeCollection BuildInTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1993]
The build in types.

Reference

ISymbolLoader Interface [▶ 2200]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.92.1.2 ISymbolLoader.Settings Property

Gets or sets the access Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolLoaderSettings Settings { get; }
```

Property Value

Type: ISymbolLoaderSettings [▶ 99]
The access method.

Reference

ISymbolLoader Interface [▶ 2200]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.92.2 ISymbolLoader Methods

The ISymbolLoader [▶ 2200] type exposes the following members.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

ISymbolLoader Interface [▶ 2200]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.93 ISymbolProvider Interface

Symbol Provider interface.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolProvider : ISymbolServer
```

The ISymbolProvider type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)

	Name	Description
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.93.1 ISymbolProvider Properties

The ISymbolProvider [▶ 2203] type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2206]	Gets the data types (Inherited from ISymbolServer [▶ 2205].)
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding. (Inherited from ISymbolServer [▶ 2205].)
	RootNamespaceName [▶ 2204]	Gets the name of the root namespace
	Symbols [▶ 2207]	Gets the symbols. (Inherited from ISymbolServer [▶ 2205].)

Reference

ISymbolProvider Interface [▶ 2203]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.93.1.1 ISymbolProvider.RootNamespaceName Property

Gets the name of the root namespace

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string RootNamespaceName { get; }
```

Property Value

Type: String
The namespace.

Reference

ISymbolProvider Interface [▶ 2203]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.93.2 ISymbolProvider Methods

The ISymbolProvider [▶ 2203] type exposes the following members.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously. (Inherited from ISymbolServer [▶ 2205].)
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously (Inherited from ISymbolServer [▶ 2205].)

Reference

ISymbolProvider Interface [▶ 2203]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.94 ISymbolServer Interface

Symbol Server Interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolServer
```

The ISymbolServer type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2206]	Gets the data types
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding.
	Symbols [▶ 2207]	Gets the symbols.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously.
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.94.1 ISymbolServer Properties

The ISymbolServer [▶ 2205] type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2206]	Gets the data types
	DefaultValueEncoding [▶ 2206]	Gets the default value encoding.
	Symbols [▶ 2207]	Gets the symbols.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.94.1.1 ISymbolServer.DataTypes Property

Gets the data types

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataTypeCollection<IDataType> DataTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].IDataType [▶ 1986].
The data types.

Remarks

This property reads the DataTypes synchronously, if the data is not available yet. For performance reasons, the asynchronous counterpart GetDataTypesAsync(CancellationToken) [▶ 2208] should be preferred for the first call.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolServer.GetDataTypesAsync(CancellationToken) [▶ 2208]

6.11.94.1.2 ISymbolServer.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.94.1.3 ISymbolServer.Symbols Property

Gets the symbols.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> Symbols { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The symbols.

Remarks

This property reads the Symbol information synchronously, if the data is not available yet. For performance reasons, the asynchronous counterpart GetSymbolsAsync(CancellationToken) [▶ 2208] should be preferred for the first call.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolServer.GetSymbolsAsync(CancellationToken) [▶ 2208]

6.11.94.2 ISymbolServer Methods

The ISymbolServer [▶ 2205] type exposes the following members.

Methods

	Name	Description
	GetDataTypesAsync [▶ 2208]	Gets the data types asynchronously.
	GetSymbolsAsync [▶ 2208]	Gets the symbols asynchronously

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.94.2.1 ISymbolServer.GetDataTypesAsync Method

Gets the data types asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultDataTypes> GetDataTypesAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultDataTypes [▶ 2347].
A task that represents the asynchronous 'GetDataTypes' operation. The ResultDataTypes [▶ 2347] parameter contains the data types (DataTypes [▶ 2349]) and the ErrorCode [▶ 992] after execution.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolServer.DataTypes [▶ 2206]

6.11.94.2.2 ISymbolServer.GetSymbolsAsync Method

Gets the symbols asynchronously

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultSymbols> GetSymbolsAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultSymbols [▶ 2353].
A task that represents the asynchronous 'GetDataTypes' operation. The ResultSymbols [▶ 2353] parameter contains the data types (Symbols [▶ 2358]) and the ErrorCode [▶ 992] after execution.

Reference

ISymbolServer Interface [▶ 2205]

TwinCAT.TypeSystem Namespace [▶ 1622]

ISymbolServer.Symbols [▶ 2207]

6.11.95 ITypeAttribute Interface

Interface for ADS attributes

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ITypeAttribute
```

The ITypeAttribute type exposes the following members.

Properties

	Name	Description
	Name [▶ 2210]	Name of the Attribute
	Value [▶ 2210]	Gets the value of the attribute

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.95.1 ITypeAttribute Properties

The ITypeAttribute [▶ 2209] type exposes the following members.

Properties

	Name	Description
	Name [▶ 2210]	Name of the Attribute
	Value [▶ 2210]	Gets the value of the attribute

Reference

ITypeAttribute Interface [▶ 2209]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.95.1.1 ITypeAttribute.Name Property

Name of the Attribute

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

ITypeAttribute Interface [▶ 2209]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.95.1.2 ITypeAttribute.Value Property

Gets the value of the attribute

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Value { get; }
```

Property Value

Type: String
The value.

Reference

ITypeAttribute Interface [▶ 2209]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96 ITypeAttributeCollection Interface

Interface ITypeAttributeCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ITypeAttributeCollection : IList<ITypeAttribute>,
    ICollection<ITypeAttribute>, IEnumerable<ITypeAttribute>, IEnumerable
```

The ITypeAttributeCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2213]	Gets the String with the specified name.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Contains(String) [▶ 2214]	Determines whether this ITypeAttributeCollection contains the specified attribute.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ITypeAttribute [▶ 2209]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)

	Name	Description
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	TryGetAttribute [▶ 2215]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2215]	Tries to get the specified Attribute value.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.1 ITypeAttributeCollection Properties

The ITypeAttributeCollection [▶ 2211] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	IsReadOnly	Gets a value indicating whether the ICollection.T. is read-only. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2213]	Gets the String with the specified name.

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.1.1 ITypeAttributeCollection.Item Property

Overload List

	Name	Description
	Item.Int32.	Gets or sets the element at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2213]	Gets the String with the specified name.

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

ITypeAttributeCollection.Item Property (String)

Gets the String with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: String
System.String.

Reference

ITypeAttributeCollection Interface [▶ 2211]

Item Overload [▶ 2212]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.2 ITypeAttributeCollection Methods

The ITypeAttributeCollection [▶ 2211] type exposes the following members.

Methods

	Name	Description
	Add	Adds an item to the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Clear	Removes all items from the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	Contains(String) [▶ 2214]	Determines whether this ITypeAttributeCollection [▶ 2211] contains the specified attribute.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	CopyTo	Copies the elements of the ICollection.T. to an Array, starting at a particular Array index. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	GetEnumerator	Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.ITypeAttribute [▶ 2209]..)
	IndexOf	Determines the index of a specific item in the IList.T.. (Inherited from IList.ITypeAttribute [▶ 2209]..)

	Name	Description
	Insert	Inserts an item to the IList.T. at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	Remove	Removes the first occurrence of a specific object from the ICollection.T.. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)
	RemoveAt	Removes the IList.T. item at the specified index. (Inherited from IList.ITypeAttribute [▶ 2209]..)
	TryGetAttribute [▶ 2215]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2215]	Tries to get the specified Attribute value.

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.2.1 ITypeAttributeCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2214]	Determines whether this ITypeAttributeCollection [▶ 2211] contains the specified attribute.
	Contains(T)	Determines whether the ICollection.T. contains a specific value. (Inherited from ICollection.ITypeAttribute [▶ 2209]..)

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

ITypeAttributeCollection.Contains Method (String)

Determines whether this ITypeAttributeCollection [▶ 2211] contains the specified attribute.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Reference

ITypeAttributeCollection Interface [▶ 2211]

Contains Overload [▶ 2214]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.2.2 ITypeAttributeCollection.TryGetAttribute Method

Tries to get the specified ITypeAttribute [▶ 2209]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetAttribute(
    string name,
    out ITypeAttribute attribute
)
```

Parameters

name	Type: System.String The name of the ITypeAttribute [▶ 2209].
attribute	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209]. The attribute.

Return Value

Type: Boolean
true if found, false otherwise.

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.96.2.3 ITypeAttributeCollection.TryGetValue Method

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryGetValue(
    string name,
    out string value
)
```

Parameters

name	Type: System.String The name.
value	Type: System.String. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

ITypeAttributeCollection Interface [▶ 2211]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97 ITypeMarshaler Interface

Interface ITypeMarshaler

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ITypeMarshaler
```

The ITypeMarshaler type exposes the following members.

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type.
	Marshal [▶ 2219]	
	MarshalSize [▶ 2219]	Gets the byte size of the value when marshalled.
	Unmarshal [▶ 2220]	

Remarks

The ITypeMarshaler is the common base interface for marshalling classes. It support marshalling / unmarshalling of primitive managed values.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97.1 ITypeMarshaler Methods

The ITypeMarshaler [▶ 2216] type exposes the following members.

Methods

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type.
	Marshal [▶ 2219]	
	MarshalSize [▶ 2219]	Gets the byte size of the value when marshalled.
	Unmarshal [▶ 2220]	

Reference

ITypeMarshaler Interface [▶ 2216]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97.1.1 ITypeMarshaler.CanMarshal Method

Overload List

	Name	Description
	CanMarshal(Object) [▶ 2217]	Determines whether ADS can marshal the specified value
	CanMarshal(Type) [▶ 2218]	Determines whether ADS can marshal the specified managed data type.

Reference

ITypeMarshaler Interface [▶ 2216]

TwinCAT.TypeSystem Namespace [▶ 1622]

ITypeMarshaler.CanMarshal Method (Object)

Determines whether ADS can marshal the specified value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool CanMarshal(
    Object value
)
```

Parameters

value	Type: System.Object The value.

Return Value

Type: Boolean
true if this instance can marshal the specified value; otherwise, false.

Reference

ITypeMarshaler Interface [▶ 2216]

CanMarshal Overload [▶ 2217]

TwinCAT.TypeSystem Namespace [▶ 1622]

ITypeMarshaler.CanMarshal Method (Type)

Determines whether ADS can marshal the specified managed data type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool CanMarshal(
    Type type
)
```

Parameters

type	Type: System.Type The Managed data type.

Return Value

Type: Boolean
true if this instance can marshal the specified managed type; otherwise, false.

Reference

ITypeMarshaler Interface [▶ 2216]

CanMarshal Overload [▶ 2217]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97.1.2 ITypeMarshaler.Marshal Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Marshal(
    Object value,
    Encoding encoding,
    Span destination
)
```

Parameters

value	Type: System.Object
encoding	Type: System.Text.Encoding
destination	Type: Span

Return Value

Type: Int32

Reference

ITypeMarshaler Interface [▶ 2216]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97.1.3 ITypeMarshaler.MarshalSize Method

Gets the byte size of the value when marshalled.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int MarshalSize(
    Object value,
    Encoding encoding
)
```

Parameters

value	Type: System.Object The value.
encoding	Type: System.Text.Encoding The encoding.

Return Value

Type: Int32
The marshal size of the value.

Reference

ITypeMarshaler Interface [▶ 2216]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.97.1.4 ITypeMarshaler.Unmarshal Method

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int Unmarshal(
    Type type,
    ReadOnlySpan source,
    void encoding,
    byte value
)
```

Parameters

type	Type: System.Type
source	Type: ReadOnlySpan
encoding	Type: System.Void
value	Type: System.Byte

Return Value

Type: Int32

Reference

ITypeMarshaler Interface [▶ 2216]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.98 IUnionInstance Interface

Interface for an Instance of the IUnionType [▶ 2224].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IUnionInstance : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IUnionInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	FieldInstances [▶ 2223]	Gets the field instances of the Union
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)

	Name	Description
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

6.11.98.1 IUnionInstance Properties

The IUnionInstance [▶ 2220] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	FieldInstances [▶ 2223]	Gets the field instances of the Union
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IUnionInstance Interface [▶ 2220]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.98.1.1 IUnionInstance.FieldInstances Property

Gets the field instances of the Union

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<ISymbol> FieldInstances { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].ISymbol [▶ 2176].
The field instances.

Reference

IUnionInstance Interface [▶ 2220]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.99 IUnionType Interface

Interface for an union data type.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IUnionType : IDataType,
    IBitSize
```

The IUnionType type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Fields [▶ 2226]	Gets a readonly collection of the Members [▶ 2040] of the IUnionType.
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IDataType [▶ 1986]

6.11.99.1 IUnionType Properties

The IUnionType [▶ 2224] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1987]	Gets the attributes of the IDataType [▶ 1986] (Inherited from IDataType [▶ 1986].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 1988]	Gets the Data Type category (Inherited from IDataType [▶ 1986].)
	Comment [▶ 1988]	Gets the comment behind the variable declaration. (Inherited from IDataType [▶ 1986].)
	Fields [▶ 2226]	Gets a readonly collection of the Members [▶ 2040] of the IUnionType [▶ 2224].
	FullName [▶ 1989]	Gets the full name of the IDataType [▶ 1986] (Namespace + Name) (Inherited from IDataType [▶ 1986].)
	Id [▶ 1989]	Gets the ID of the DataType (Inherited from IDataType [▶ 1986].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a container type (Inherited from IDataType [▶ 1986].)
	IsPointer [▶ 1990]	Gets a value indicating whether this IDataType [▶ 1986] is a pointer type (Inherited from IDataType [▶ 1986].)
	IsPrimitive [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is primitive (Inherited from IDataType [▶ 1986].)
	IsReference [▶ 1991]	Gets a value indicating whether this IDataType [▶ 1986] is a reference type (Inherited from IDataType [▶ 1986].)
	Name [▶ 1992]	Gets the name of the Data Type (without namespace) (Inherited from IDataType [▶ 1986].)
	Namespace [▶ 1992]	Gets the namespace string within the IDataType [▶ 1986] exists. (Inherited from IDataType [▶ 1986].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)

Reference

IUnionType Interface [▶ 2224]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.99.1.1 IUnionType.Fields Property

Gets a readonly collection of the Members [▶ 2040] of the IUnionType [▶ 2224].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IFieldCollection Fields { get; }
```

Property Value

Type: IFieldCollection [▶ 2042]
The members as readonly collection.

Remarks

If the IStructType [▶ 2162] is derived, only the extended members are returned. To get all supported members down the inheritance chain, use the AllMembers [▶ 2165] property.

Reference

IUnionType Interface [▶ 2224]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100 IValue Interface

Symbol Value Interface

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValue
```

The IValue type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data.
	DataType [▶ 2228]	Gets the data type bound to this IValue
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue is a primitive value.

	Name	Description
	Symbol [▶ 2229]	Gets the symbol bound to this IValue.
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet)

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS)
	ReadAsync [▶ 2231]	Reads the value (via ADS)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value.
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value.
	Write [▶ 2233]	Writes the value (via ADS)
	WriteAsync [▶ 2233]	Writes the value (via ADS)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1 IValue Properties

The IValue [▶ 2226] type exposes the following members.

Properties

	Name	Description
	Age [▶ 2228]	Gets the age of the value (last successful read of the value)
	CachedRaw [▶ 2228]	Gets the cached Raw internal Data.
	DataType [▶ 2228]	Gets the data type bound to this IValue [▶ 2226]
	IsPrimitive [▶ 2229]	Gets a value indicating whether this IValue [▶ 2226] is a primitive value.
	Symbol [▶ 2229]	Gets the symbol bound to this IValue [▶ 2226].
	TimeStamp [▶ 2230]	Gets the Time stamp of the last successful read of the Value (local user time, UTC)
	UpdateMode [▶ 2230]	Gets the update mode (not implemented yet)

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.1 IValue.Age Property

Gets the age of the value (last successful read of the value)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
TimeSpan Age { get; }
```

Property Value

Type: TimeSpan
The age.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValue.TimeStamp [▶ 2230]

6.11.100.1.2 IValue.CachedRaw Property

Gets the cached Raw internal Data.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
byte[] CachedRaw { get; }
```

Property Value

Type: .Byte.
The raw cached data.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.3 IValue.DataType Property

Gets the data type bound to this IValue [▶ 2226]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataType DataType { get; }
```

Property Value

Type: IDataType [▶ 1986]
The type of the data.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.4 IValue.IsPrimitive Property

Gets a value indicating whether this IValue [▶ 2226] is a primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool IsPrimitive { get; }
```

Property Value

Type: Boolean
true if this instance is primitive; otherwise, false.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.5 IValue.Symbol Property

Gets the symbol bound to this IValue [▶ 2226].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbol Symbol { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The symbol.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.6 IValue.TimeStamp Property

Gets the Time stamp of the last successful read of the Value (local user time, UTC)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
DateTimeOffset TimeStamp { get; }
```

Property Value

Type: DateTimeOffset
The read time stamp.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.1.7 IValue.UpdateMode Property

Gets the update mode (not implemented yet)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ValueUpdateMode UpdateMode { get; }
```

Property Value

Type: ValueUpdateMode [▶ 150]
The update mode.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2 IValue Methods

The IValue [▶ 2226] type exposes the following members.

Methods

	Name	Description
	Read [▶ 2231]	Reads the value (via ADS)
	ReadAsync [▶ 2231]	Reads the value (via ADS)
	ResolveValue [▶ 2232]	Resolves the Value object to its primitive value.
	TryResolveValue [▶ 2232]	Tries to resolves the Value object to its primitive value.
	Write [▶ 2233]	Writes the value (via ADS)
	WriteAsync [▶ 2233]	Writes the value (via ADS)

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.1 IValue.Read Method

Reads the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Read()
```

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.2 IValue.ReadAsync Method

Reads the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultAccess> ReadAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultAccess [▶ 2556].
Task<ReadValueResult>.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.3 IValue.ResolveValue Method

Resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ResolveValue(
    bool resolveEnumToPrimitive
)
```

Parameters

resolveEnumToPrimitive	Type: System.Boolean if set to trueIEnumValue [▶ 2028]s are resolved to their primitives also.

Return Value

Type: Object
System.Object.

Remarks

If the value is not primitive, this method returns the IValue [▶ 2226] itself.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.4 IValue.TryResolveValue Method

Tries to resolves the Value object to its primitive value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool TryResolveValue(
    bool resolveEnumToPrimitive,
    out Object value
)
```

Parameters

resolveEnumToPrimitive	Type: System.Boolean if set to trueIEnumValue [▶ 2028]s are resolved to their primitives also.
value	Type: System.Object. The value.

Return Value

Type: Boolean
true if value can be resolved, false otherwise.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.5 IValue.Write Method

Writes the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void Write()
```

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.100.2.6 IValue.WriteAsync Method

Writes the value (via ADS)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWriteAccess> WriteAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
Task<WriteValueResult>.

Reference

IValue Interface [▶ 2226]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.101 IValueAccessorProvider Interface

Interface IValueAccessorProvider

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValueAccessorProvider
```

The IValueAccessorProvider type exposes the following members.

Properties

	Name	Description
	ValueAccessor [▶ 2235]	Gets the value accessor.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.101.1 IValueAccessorProvider Properties

The IValueAccessorProvider [▶ 2234] type exposes the following members.

Properties

	Name	Description
	ValueAccessor [▶ 2235]	Gets the value accessor.

Reference

IValueAccessorProvider Interface [▶ 2234]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.101.1.1 IValueAccessorProvider.ValueAccessor Property

Gets the value accessor.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IAccessorRawValue ValueAccessor { get; }
```

Property Value

Type: IAccessorRawValue [▶ 2544]
The value accessor.

Reference

IValueAccessorProvider Interface [▶ 2234]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.102 IValueAnySymbol Interface

Interface IValueAnySymbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValueAnySymbol : ISymbol,
    IAttributedInstance, IInstance, IBitSize
```

The IValueAnySymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)

	Name	Description
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	ReadAnyValue(Type) [▶ 2239]	Reads the value of this Value into a new created instance of the managed type
	ReadAnyValue(Type, Int32) [▶ 2240]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type
	ReadAnyValueAsync [▶ 2241]	Reads the (AnyType) value asynchronously.
	UpdateAnyValue(Object.) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteAnyValue(Object) [▶ 2243]	Writes the value represented by the managed value to this Value [▶ 2254]
	WriteAnyValue(Object, Int32) [▶ 2244]	Writes the value represented by the managed value to this Value [▶ 2254]

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.ISymbol [▶ 2176]

6.11.102.1 IValueAnySymbol Properties

The IValueAnySymbol [▶ 2235] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)

	Name	Description
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.102.2 IValueAnySymbol Methods

The IValueAnySymbol [▶ 2235] type exposes the following members.

Methods

	Name	Description
	ReadAnyValue(Type) [▶ 2239]	Reads the value of this Value [▶ 2235] into a new created instance of the managed type
	ReadAnyValue(Type, Int32) [▶ 2240]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type

	Name	Description
	ReadAnyValueAsync [▶ 2241]	Reads the (AnyType) value asynchronously.
	UpdateAnyValue(Object.) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.
	WriteAnyValue(Object) [▶ 2243]	Writes the value represented by the managed value to this Value [▶ 2254]
	WriteAnyValue(Object, Int32) [▶ 2244]	Writes the value represented by the managed value to this Value [▶ 2254]

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.102.2.1 IValueAnySymbol.ReadAnyValue Method

Overload List

	Name	Description
	ReadAnyValue(Type) [▶ 2239]	Reads the value of this Value [▶ 2235] into a new created instance of the managed type
	ReadAnyValue(Type, Int32) [▶ 2240]	Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue Method (Type)

Reads the value of this Value [▶ 2235] into a new created instance of the managed type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAnyValue(
    Type managedType
)
```

Parameters

managedType	Type: System.Type The tp.

Return Value

Type: Object
Read value (System.Object).

Reference

IValueAnySymbol Interface [▶ 2235]

ReadAnyValue Overload [▶ 2239]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

IValueAnySymbol.ReadAnyValue Method (Type, Int32)

Reads the value of this Value [▶ 2254] into a new created instance of the managed type

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadAnyValue(
    Type managedType,
    int timeout
)
```

Parameters

managedType	Type: System.Type The tp.
timeout	Type: System.Int32 The timeout in ms.

Return Value

Type: Object
Read value (System.Object).

Reference

IValueAnySymbol Interface [▶ 2235]

ReadAnyValue Overload [▶ 2239]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

6.11.102.2.2 IValueAnySymbol.ReadAnyValueAsync Method

Reads the (AnyType) value asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadValueAccess> ReadAnyValueAsync(
    Type managedType,
    CancellationToken cancel
)
```

Parameters

managedType	Type: System.Type Managed type of the value to read.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
A task object that is representing the asynchronous 'ReadAnyValue' operation. The result will be returned in a ResultReadValueAccess [▶ 2566], which contains the Value [▶ 2570] and the ErrorCode [▶ 2559].

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.102.2.3 IValueAnySymbol.UpdateAnyValue Method

Overload List

	Name	Description
	UpdateAnyValue(Object.) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.
	UpdateAnyValue(Object., Int32) [▶ 2242]	Reads the value of this Value [▶ 2254] into the specified managed value.

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.UpdateAnyValue Method (Object.)

Reads the value of this Value [▶ 2254] into the specified managed value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void UpdateAnyValue(
    ref Object managedObject
)
```

Parameters

managedObject	Type: System.Object. The managed object.

Return Value

Type:
Read value (System.Object).

Reference

IValueAnySymbol Interface [▶ 2235]

UpdateAnyValue Overload [▶ 2241]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

IValueAnySymbol.UpdateAnyValue Method (Object., Int32)

Reads the value of this Value [▶ 2254] into the specified managed value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void UpdateAnyValue(
    ref Object managedObject,
    int timeout
)
```

Parameters

managedObject	Type: System.Object. The managed object.
timeout	Type: System.Int32 The timeout.

Return Value

Type:
Read value (System.Object).

Reference

IValueAnySymbol Interface [▶ 2235]

UpdateAnyValue Overload [▶ 2241]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.WriteAnyValue(Object) [▶ 2243]

6.11.102.2.4 IValueAnySymbol.WriteAnyValue Method

Overload List

	Name	Description
	WriteAnyValue(Object) [▶ 2243]	Writes the value represented by the managed value to this Value [▶ 2254]
	WriteAnyValue(Object, Int32) [▶ 2244]	Writes the value represented by the managed value to this Value [▶ 2254]

Reference

IValueAnySymbol Interface [▶ 2235]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.WriteAnyValue Method (Object)

Writes the value represented by the managed value to this Value [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAnyValue(
    Object managedValue
)
```

Parameters

managedValue	Type: System.Object The managed value.

Reference

IValueAnySymbol Interface [▶ 2235]

WriteAnyValue Overload [▶ 2243]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

IValueAnySymbol.WriteAnyValue Method (Object, Int32)

Writes the value represented by the managed value to this Value [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteAnyValue(
    Object managedValue,
    int timeout
)
```

Parameters

managedValue	Type: System.Object The managed value.
timeout	Type: System.Int32 The timeout in ms.

Reference

IValueAnySymbol Interface [▶ 2235]

WriteAnyValue Overload [▶ 2243]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueAnySymbol.ReadAnyValue(Type) [▶ 2239]

IValueAnySymbol.UpdateAnyValue(Object.) [▶ 2242]

6.11.103 IValueRawSymbol Interface

Interface IValueRawSymbol

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValueRawSymbol : IHierarchicalSymbol,
    ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IValueRawSymbol type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasValue [▶ 2248]	Gets a value indicating whether this IValueSymbol [▶ 2254] has a value.
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)

	Name	Description
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueAccessor [▶ 2248]	Gets the value accessor.
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	ReadRawValue. [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValueAsync [▶ 2251]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.
	SetParent [▶ 2051]	Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [▶ 2048].)
	WriteRawValue(.Byte.) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 2253]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Events

	Name	Description
	RawValueChanged [▶ 2254]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IHierarchicalSymbol [▶ 2048]

6.11.103.1 IValueRawSymbol Properties

The IValueRawSymbol [▶ 2244] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasValue [▶ 2248]	Gets a value indicating whether this IValueSymbol [▶ 2254] has a value.
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)

	Name	Description
	ValueAccessor [▶ 2248]	Gets the value accessor.
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.1.1 IValueRawSymbol.HasValue Property

Gets a value indicating whether this IValueSymbol [▶ 2254] has a value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool HasValue { get; }
```

Property Value

Type: Boolean
true if this instance has value; otherwise, false.

Remarks

A VirtualSymbol does not support values, but in terms of the IValueSymbol [▶ 2254] definition, is a IValueSymbol [▶ 2254]

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.1.2 IValueRawSymbol.ValueAccessor Property

Gets the value accessor.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IAccessorRawValue ValueAccessor { get; }
```

Property Value

Type: IAccessorRawValue [▶ 2544]
The value accessor.

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.2 IValueRawSymbol Methods

The IValueRawSymbol [▶ 2244] type exposes the following members.

Methods

	Name	Description
	ReadRawValue. [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValueAsync [▶ 2251]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.
	SetParent [▶ 2051]	Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [▶ 2048].)
	WriteRawValue(.Byte.) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValueAsync [▶ 2253]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.2.1 IValueRawSymbol.ReadRawValue Method

Overload List

	Name	Description
	ReadRawValue. [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	ReadRawValue(Int32) [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueRawSymbol.ReadRawValue Method

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
byte[] ReadRawValue()
```

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Reference

IValueRawSymbol Interface [▶ 2244]

ReadRawValue Overload [▶ 2249]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueRawSymbol.ReadRawValue Method (Int32)

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
byte[] ReadRawValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.

Field Value

Type: .Byte.
The raw value.

Return Value

Type: .Byte.
System.Byte[].

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueRawSymbol Interface [▶ 2244]

ReadRawValue Overload [▶ 2249]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.2.2 IValueRawSymbol.ReadRawValueAsync Method

Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadRawAccess> ReadRawValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Field Value

Type: Task.ResultReadRawAccess [▶ 2564].
The raw value.

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].
System.Byte[].

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.2.3 IValueRawSymbol.WriteRawValue Method

Overload List

	Name	Description
	WriteRawValue(.Byte.) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)
	WriteRawValue(.Byte., Int32) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueRawSymbol.WriteRawValue Method (.Byte.)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteRawValue(
    byte[] rawValue
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.

Reference

IValueRawSymbol Interface [▶ 2244]

WriteRawValue Overload [▶ 2252]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueRawSymbol.WriteRawValue Method (.Byte., Int32)

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteRawValue(
    byte[] rawValue,
    int timeout
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.
timeout	Type: System.Int32 The timeout.

Field Value

Type:
The value.

Remarks

A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueRawSymbol Interface [▶ 2244]

WriteRawValue Overload [▶ 2252]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.2.4 IValueRawSymbol.WriteRawValueAsync Method

Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWriteAccess> WriteRawValueAsync(
    byte[] rawValue,
    CancellationToken cancel
)
```

Parameters

rawValue	Type: .System.Byte. The value as byte array.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A task that represents the asynchronous read operation. The ResultRead [▶ 1008] parameter contains the total number of bytes read into the buffer (ReadBytes [▶ 1010]) and the ErrorCode [▶ 992] after execution..

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.3 IValueRawSymbol Events

The IValueRawSymbol [▶ 2244] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 2254]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.103.3.1 IValueRawSymbol.RawValueChanged Event

Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<RawValueChangedEventArgs> RawValueChanged
```

Value

Type: System.EventHandler.RawValueChangedEventArgs [▶ 2289].

Reference

IValueRawSymbol Interface [▶ 2244]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104 IValueSymbol Interface

Interface for a ISymbol [▶ 2176] that supports values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IValueSymbol : IValueRawSymbol,
    IHierarchicalSymbol, ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IValueSymbol type exposes the following members.

Properties

	Name	Description
	AccessRights [▶ 2259]	Gets the access rights.
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	Connection [▶ 2260]	Gets the connection that produces values for this IValueSymbol
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasValue [▶ 2248]	Gets a value indicating whether this IValueSymbol has a value. (Inherited from IValueRawSymbol [▶ 2244].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	NotificationSettings [▶ 2260]	Gets or sets the notification settings.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueAccessor [▶ 2248]	Gets the value accessor. (Inherited from IValueRawSymbol [▶ 2244].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	ReadRawValue. [▶ 2250]	Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	ReadRawValue(Int32) [▶ 2250]	Reads the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	ReadRawValueAsync [▶ 2251]	Reads the raw value of the IValueSymbol (Ads Read / Write) asynchronously. (Inherited from IValueRawSymbol [▶ 2244].)
	ReadValue. [▶ 2263]	Reads the Value of the IValueSymbol
	ReadValue(Int32) [▶ 2263]	Reads the Value of the IValueSymbol
	ReadValueAsync [▶ 2264]	Reads the Value of the IValueSymbol asynchronously.
	SetParent [▶ 2051]	Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [▶ 2048].)
	TryReadValue [▶ 2265]	Reads the Value of the IValueSymbol
	TryWriteValue [▶ 2265]	Writes the Value of the IValueSymbol
	WriteRawValue(.Byte.) [▶ 2252]	Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	WriteRawValue(.Byte., Int32) [▶ 2252]	Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)

	Name	Description
	WriteRawValueAsync [▶ 2253]	Writes the raw value of the IValueSymbol (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	WriteValue(Object) [▶ 2266]	Writes the Value of the IValueSymbol
	WriteValue(Object, Int32) [▶ 2267]	Writes the Value of the IValueSymbol
	WriteValueAsync [▶ 2268]	Writes the Value of the IValueSymbol

Events

	Name	Description
	RawValueChanged [▶ 2254]	Occurs when the RawValue of the IValueSymbol has changed. (Inherited from IValueRawSymbol [▶ 2244].)
	ValueChanged [▶ 2269]	Occurs when the (Primitive) value of the IValueSymbol has changed.

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol to an observable sequence of values and writes them to the IValueSymbol. (Defined by ValueSymbolExtensions [▶ 1106].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IValueRawSymbol [▶ 2244]

6.11.104.1 IValueSymbol Properties

The IValueSymbol [▶ 2254] type exposes the following members.

Properties

	Name	Description
	AccessRights [▶ 2259]	Gets the access rights.
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	Connection [▶ 2260]	Gets the connection that produces values for this IValueSymbol [▶ 2254]
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasValue [▶ 2248]	Gets a value indicating whether this IValueSymbol [▶ 2254] has a value. (Inherited from IValueRawSymbol [▶ 2244].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)

	Name	Description
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	NotificationSettings [▶ 2260]	Gets or sets the notification settings.
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueAccessor [▶ 2248]	Gets the value accessor. (Inherited from IValueRawSymbol [▶ 2244].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.1.1 IValueSymbol.AccessRights Property

Gets the access rights.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
SymbolAccessRights AccessRights { get; }
```

Property Value

Type: SymbolAccessRights [▶ 2396]
The access rights.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.1.2 IValueSymbol.Connection Property

Gets the connection that produces values for this IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IConnection Connection { get; }
```

Property Value

Type: IConnection [▶ 74]
The connection object.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.1.3 IValueSymbol.NotificationSettings Property

Gets or sets the notification settings.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
INotificationSettings NotificationSettings { get; set; }
```

Property Value

Type: INotificationSettings [▶ 972]
The notification settings.

Remarks

The NotificationSettings will be inherited from Parent [▶ 2182] if the setting is not overwritten.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2 IValueSymbol Methods

The IValueSymbol [▶ 2254] type exposes the following members.

Methods

	Name	Description
	ReadRawValue. [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	ReadRawValue(Int32) [▶ 2250]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	ReadRawValueAsync [▶ 2251]	Reads the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) asynchronously. (Inherited from IValueRawSymbol [▶ 2244].)
	ReadValue. [▶ 2263]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValue(Int32) [▶ 2263]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValueAsync [▶ 2264]	Reads the Value of the IValueSymbol [▶ 2254] asynchronously.
	SetParent [▶ 2051]	Sets the parent of the Symbol (Inherited from IHierarchicalSymbol [▶ 2048].)
	TryReadValue [▶ 2265]	Reads the Value of the IValueSymbol [▶ 2254]
	TryWriteValue [▶ 2265]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteRawValue(.Byte.) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	WriteRawValue(.Byte., Int32) [▶ 2252]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	WriteRawValueAsync [▶ 2253]	Writes the raw value of the IValueSymbol [▶ 2254] (Ads Read / Write) (Inherited from IValueRawSymbol [▶ 2244].)
	WriteValue(Object) [▶ 2266]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValue(Object, Int32) [▶ 2267]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValueAsync [▶ 2268]	Writes the Value of the IValueSymbol [▶ 2254]

Extension Methods

	Name	Description
	PollValuesAnnotated(IObservable.Unit.) [▶ 1111]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence annotated value on trigger sequence (Defined by ValueSymbolExtensions [▶ 1106].)

	Name	Description
	PollValuesAnnotated(TimeSpan) [▶ 1112]	Overloaded. Polls the values as ValueChangedEventArgs [▶ 2439] sequence with a specified period time. (Defined by ValueSymbolExtensions [▶ 1106].)
	WhenValueChanged [▶ 1113]	Gets an observable sequence when the value of the IValueSymbol [▶ 2254] has changed. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object.) [▶ 1117]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception.) [▶ 1118]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., CancellationToken) [▶ 1119]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)
	WriteValues(IObservable.Object., Action.Exception., CancellationToken) [▶ 1120]	Overloaded. Subscribes the IValueSymbol [▶ 2254] to an observable sequence of values and writes them to the IValueSymbol [▶ 2254]. (Defined by ValueSymbolExtensions [▶ 1106].)

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.1 IValueSymbol.ReadValue Method

Overload List

	Name	Description
	ReadValue. [▶ 2263]	Reads the Value of the IValueSymbol [▶ 2254]
	ReadValue(Int32) [▶ 2263]	Reads the Value of the IValueSymbol [▶ 2254]

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueSymbol.ReadValue Method

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadValue()
```

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

IValueSymbol Interface [▶ 2254]

ReadValue Overload [▶ 2262]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueSymbol.ReadValue Method (Int32)

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object ReadValue(
    int timeout
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.

Field Value

Type: Object
The value.

Return Value

Type: Object
System.Object.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface [▶ 2254]

ReadValue Overload [▶ 2262]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.2 IValueSymbol.ReadValueAsync Method

Reads the Value of the IValueSymbol [▶ 2254] asynchronously.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadValueAccess> ReadValueAsync(
    CancellationToken cancel
)
```

Parameters

cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultReadValueAccess [▶ 2566].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultReadValueAccess [▶ 2566] return value and contains the Value [▶ 2570] and the ErrorCode [▶ 2559].

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.3 IValueSymbol.TryReadValue Method

Reads the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryReadValue(
    int timeout,
    out Object value
)
```

Parameters

timeout	Type: System.Int32 The timeout in ms.
value	Type: System.Object. The symbol value.

Return Value

Type: Int32
The error code.

Remarks

Calling on primitive types, a call of this method will return the primitive value. On complex types (structures and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: the raw byte Array will be returned, in dynamic mode: A Value will be created on the fly. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.4 IValueSymbol.TryWriteValue Method

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryWriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Return Value

Type: Int32
The error code.

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.5 IValueSymbol.WriteValue Method

Overload List

	Name	Description
	WriteValue(Object) [▶ 2266]	Writes the Value of the IValueSymbol [▶ 2254]
	WriteValue(Object, Int32) [▶ 2267]	Writes the Value of the IValueSymbol [▶ 2254]

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueSymbol.WriteValue Method (Object)

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue(
    Object value
)
```

Parameters

value	Type: System.Object The value.

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

IValueSymbol Interface [▶ 2254]

WriteValue Overload [▶ 2266]

TwinCAT.TypeSystem Namespace [▶ 1622]

IValueSymbol.WriteValue Method (Object, Int32)

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
void WriteValue(
    Object value,
    int timeout
)
```

Parameters

value	Type: System.Object The value.
timeout	Type: System.Int32 The timeout in ms.

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also. A negative timeout indicates that the Default Timeout for the communication will be used. 0 means timeout is switched off.

Reference

IValueSymbol Interface [▶ 2254]

WriteValue Overload [▶ 2266]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.2.6 IValueSymbol.WriteValueAsync Method

Writes the Value of the IValueSymbol [▶ 2254]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWriteAccess> WriteValueAsync(
    Object value,
    CancellationToken cancel
)
```

Parameters

value	Type: System.Object The value.
cancel	Type: System.Threading.CancellationToken The cancellation token.

Return Value

Type: Task.ResultWriteAccess [▶ 2575].
A tasks that represents the asynchronous 'ReadValue' operation. The read result is stored in the ResultWriteAccess [▶ 2575] return value and contains the ErrorCode [▶ 2559].

Remarks

Calling on primitive types, a call of this method will directly write this Value. On complex types (structs and arrays) it depends on the ISymbolLoader [▶ 2200] settings what will happen. In non dynamic modes: Only byte Arrays (of correct size) can be written) in dynamic mode: A Value that represents the value will be accepted also.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.3 IValueSymbol Events

The IValueSymbol [▶ 2254] type exposes the following members.

Events

	Name	Description
	RawValueChanged [▶ 2254]	Occurs when the RawValue of the IValueSymbol [▶ 2254] has changed. (Inherited from IValueRawSymbol [▶ 2244].)
	ValueChanged [▶ 2269]	Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.104.3.1 IValueSymbol.ValueChanged Event

Occurs when the (Primitive) value of the IValueSymbol [▶ 2254] has changed.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
event EventHandler<ValueChangedEventArgs> ValueChanged
```

Value

Type: System.EventHandler.ValueChangedEventArgs [▶ 2439].

Reference

IValueSymbol Interface [▶ 2254]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.105 IVirtualStructInstance Interface

Virtual Struct instance interface.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IVirtualStructInstance : IStructInstance,
    ISymbol, IAttributedInstance, IInstance, IBitSize
```

The IVirtualStructInstance type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [▶ 2158].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from IStructInstance [▶ 2158].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)

	Name	Description
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Methods

	Name	Description
	AddMember [▶ 2273]	Adds the member.

Remarks

Virtual struct instance are used to create a TreeView from the flat list of symbols.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.IStructInstance [▶ 2158]

6.11.105.1 IVirtualStructInstance Properties

The IVirtualStructInstance [▶ 2269] type exposes the following members.

Properties

	Name	Description
	Attributes [▶ 1982]	Gets the Type Attributes. (Inherited from IAttributedInstance [▶ 1980].)
	BitSize [▶ 1984]	Gets the size of the IDataType [▶ 1986] in bits. (Inherited from IBitSize [▶ 1982].)
	ByteSize [▶ 1984]	Gets the (aligned) size of of the Type/Instance in Bytes (Inherited from IBitSize [▶ 1982].)
	Category [▶ 2179]	Gets the Symbol/Datatype Category (Inherited from ISymbol [▶ 2176].)
	Comment [▶ 2053]	Gets the comment of the IInstance [▶ 2052] (Inherited from IInstance [▶ 2052].)
	DataType [▶ 2054]	Gets the IDataType [▶ 1986] of the IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	HasRpcMethods [▶ 2161]	Gets a value indicating whether this instance has RPC methods (Inherited from IStructInstance [▶ 2158].)
	InstanceName [▶ 2054]	Gets the name of the instance (without periods (.) (Inherited from IInstance [▶ 2052].)
	InstancePath [▶ 2055]	Gets the relative / absolute access path to the instance (with periods (.)) (Inherited from IInstance [▶ 2052].)
	IsBitType [▶ 1984]	Gets a value indicating whether this instance is not basing on a full DataType but instead of some sort of bit mapping (Inherited from IBitSize [▶ 1982].)
	IsByteAligned [▶ 1985]	Indicates that the Size of the Object is Byte aligned (BitSize % 8 == 0) (Inherited from IBitSize [▶ 1982].)

	Name	Description
	IsContainerType [▶ 2179]	Gets a value indicating whether this Symbol is acontainer type. (Inherited from ISymbol [▶ 2176].)
	IsPersistent [▶ 2180]	Gets a value indicating whether this ISymbol [▶ 2176] is persistent. (Inherited from ISymbol [▶ 2176].)
	IsPointer [▶ 2055]	Indicates that the IInstance [▶ 2052] represents a Pointer type (Pointer TO) (Inherited from IInstance [▶ 2052].)
	IsPrimitiveType [▶ 2180]	Gets a value indicating whether this instance is a primitive type. (Inherited from ISymbol [▶ 2176].)
	IsReadOnly [▶ 2181]	Indicates that this instance is read only. (Inherited from ISymbol [▶ 2176].)
	IsRecursive [▶ 2181]	Gets a value indicating whether this instance is recursive. (Inherited from ISymbol [▶ 2176].)
	IsReference [▶ 2056]	Indicates that the IInstance [▶ 2052] represents a Reference type (REFERENCE TO) (Inherited from IInstance [▶ 2052].)
	IsStatic [▶ 2056]	Gets a value indicating whether this IInstance [▶ 2052] is static. (Inherited from IInstance [▶ 2052].)
	MemberInstances [▶ 2162]	Gets the member instances of the Struct Instance [▶ 2158]. (Inherited from IStructInstance [▶ 2158].)
	Parent [▶ 2182]	Gets the parent Symbol (Inherited from ISymbol [▶ 2176].)
	Size [▶ 1985]	Gets the size of the object in bytes or Bits dependant on IsBitType [▶ 1984] (Inherited from IBitSize [▶ 1982].)
	SubSymbols [▶ 2182]	Gets the SubSymbols of the ISymbol [▶ 2176] (Inherited from ISymbol [▶ 2176].)
	TypeName [▶ 2056]	Gets the name of the DataType [▶ 1986] that is used for this IInstance [▶ 2052]. (Inherited from IInstance [▶ 2052].)
	ValueEncoding [▶ 1982]	Gets the value encoding. (Inherited from IAttributedInstance [▶ 1980].)

Reference

IVirtualStructInstance Interface [▶ 2269]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.105.2 IVirtualStructInstance Methods

The IVirtualStructInstance [▶ 2269] type exposes the following members.

Methods

	Name	Description
	AddMember [▶ 2273]	Adds the member.

Reference

IVirtualStructInstance Interface [▶ 2269]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.105.2.1 IVirtualStructInstance.AddMember Method

Adds the member.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
bool AddMember(
    ISymbol memberInstance,
    IVirtualStructInstance parent
)
```

Parameters

memberInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The member instance.
parent	Type: TwinCAT.TypeSystem.IVirtualStructInstance [▶ 2269] The parent struct instance. Usually the this pointer.

Return Value

Type: Boolean

Reference

IVirtualStructInstance Interface [▶ 2269]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106 MarshalException Class

Common Marshalling Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.MarshalException
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class MarshalException : AdsException
```

The MarshalException type exposes the following members.

Constructors

	Name	Description
	MarshalException. [▶ 2276]	Initializes a new instance of the MarshalException class.
	MarshalException(String) [▶ 2276]	Initializes a new instance of the MarshalException class.
	MarshalException(IDataType) [▶ 2277]	Initializes a new instance of the MarshalException class.
	MarshalException(String, Exception) [▶ 2277]	Initializes a new Instance of the AdsException class.
	MarshalException(IDataType, Object) [▶ 2278]	Initializes a new instance of the MarshalException class.
	MarshalException(IDataType, Type) [▶ 2278]	Initializes a new instance of the MarshalException class.
	MarshalException(IDataType, Type, String) [▶ 2279]	Initializes a new instance of the MarshalException class.
	MarshalException(IInstance, Type, MemberInfo) [▶ 2279]	Initializes a new instance of the MarshalException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1 MarshalException Constructor

Overload List

	Name	Description
	MarshalException. [▶ 2276]	Initializes a new instance of the MarshalException [▶ 2273] class.
	MarshalException(String) [▶ 2276]	Initializes a new instance of the MarshalException [▶ 2273] class.
	MarshalException(IDataType) [▶ 2277]	Initializes a new instance of the MarshalException [▶ 2273] class.
	MarshalException(String, Exception) [▶ 2277]	Initializes a new Instance of the AdsException class.
	MarshalException(IDataType, Object) [▶ 2278]	Initializes a new instance of the MarshalException [▶ 2273] class.
	MarshalException(IDataType, Type) [▶ 2278]	Initializes a new instance of the MarshalException [▶ 2273] class.
	MarshalException(IDataType, Type, String) [▶ 2279]	Initializes a new instance of the MarshalException [▶ 2273] class.

	Name	Description
	MarshalException(IInstance, Type, MemberInfo) [▶ 2279]	Initializes a new instance of the MarshalException [▶ 2273] class.

Reference

MarshalException Class [▶ 2273]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.1 MarshalException Constructor

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException()
```

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.2 MarshalException Constructor (String)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    string message
)
```

Parameters

message	Type: System.String The message.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.3 MarshalException Constructor (IDataType)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    IDataType source
)
```

Parameters

source	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] The source dataType.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.4 MarshalException Constructor (String, Exception)

Initializes a new Instance of the AdsException class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    string message,
    Exception innerException
)
```

Parameters

message	Type: System.String The message.
innerException	Type: System.Exception The inner exception.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.5 MarshalException Constructor (IDataType, Object)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    IDataType target,
    Object value
)
```

Parameters

target	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] The datatype information.
value	Type: System.Object The value.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.6 MarshalException Constructor (IDataType, Type)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    IDataType source,
    Type target
)
```

Parameters

source	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] The source.
target	Type: System.Type The target.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.7 MarshalException Constructor (IDataType, Type, String)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    IDataType source,
    Type target,
    string message
)
```

Parameters

source	Type: TwinCAT.TypeSystem.IDataType [▶ 1986] The source.
target	Type: System.Type The target.
message	Type: System.String The message.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.1.8 MarshalException Constructor (IInstance, Type, MemberInfo)

Initializes a new instance of the MarshalException [▶ 2273] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MarshalException(
    IInstance instance,
    Type tp,
    MemberInfo member
)
```

Parameters

instance	Type: TwinCAT.TypeSystem.IInstance [▶ 2052] The instance.
tp	Type: System.Type The type.
member	Type: System.Reflection.MemberInfo The member.

Reference

MarshalException Class [▶ 2273]

MarshalException Overload [▶ 2275]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.2 MarshalException Properties

The MarshalException [▶ 2273] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

MarshalException Class [▶ 2273]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.106.3 MarshalException Methods

The MarshalException [▶ 2273] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

MarshalException Class [▶ 2273]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107 MemberCollection Class

Collection of IMember [▶ 2065] objects.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection [▶ 2460].IMember [▶ 2065].
    TwinCAT.TypeSystem.MemberCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class MemberCollection : InstanceCollection<IMember>,
    IMemberCollection, IInstanceCollection<IMember>, IList<IMember>,
    ICollection<IMember>, IEnumerable<IMember>, IEnumerable
```

The MemberCollection type exposes the following members.

Constructors

	Name	Description
	MemberCollection. [▶ 2283]	Initializes a new instance of the MemberCollection class.
	MemberCollection(IEnumerable.IMember.) [▶ 2284]	Initializes a new instance of the MemberCollection class (copy constructor)

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the LIst of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2286]	Returns a read only copy of this collection (shallow copy)
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2287]	Clones this MemberCollection.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2287]	Returns an Empty Member Collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetMember [▶ 2287]	Tries to get the specified member

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.1 MemberCollection Constructor

Overload List

	Name	Description
	MemberCollection. [▶ 2283]	Initializes a new instance of the MemberCollection [▶ 2281] class.
	MemberCollection(IEnumerable.IMember.) [▶ 2284]	Initializes a new instance of the MemberCollection [▶ 2281] class (copy constructor)

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.1.1 MemberCollection Constructor

Initializes a new instance of the MemberCollection [▶ 2281] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MemberCollection()
```

Reference

MemberCollection Class [▶ 2281]

MemberCollection Overload [▶ 2283]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.1.2 MemberCollection Constructor (IEnumerable.IMember.)

Initializes a new instance of the MemberCollection [▶ 2281] class (copy constructor)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MemberCollection(
    IEnumerable<IMember> coll
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.IMember [▶ 2065]. The coll.

Reference

MemberCollection Class [▶ 2281]

MemberCollection Overload [▶ 2283]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.2 MemberCollection Properties

The MemberCollection [▶ 2281] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the IList of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.3 MemberCollection Methods

The MemberCollection [▶ 2281] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2286]	Returns a read only copy of this collection (shallow copy)
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2287]	Clones this MemberCollection [▶ 2281].
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2287]	Returns an Empty Member Collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetMember [▶ 2287]	Tries to get the specified member

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.3.1 MemberCollection.AsReadOnly Method

Returns a read only copy of this collection (shallow copy)

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemberCollection AsReadOnly()
```

Return Value

Type: ReadOnlyMemberCollection [▶ 2322]
The readonly copy.

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.3.2 MemberCollection.Clone Method

Clones this MemberCollection [▶ 2281].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public MemberCollection Clone()
```

Return Value

Type: MemberCollection [▶ 2281]
A cloned MemberCollection [▶ 2281].

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.3.3 MemberCollection.Empty Method

Returns an Empty Member Collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static MemberCollection Empty()
```

Return Value

Type: MemberCollection [▶ 2281]
MemberCollection.

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.107.3.4 MemberCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMember(
    string fieldName,
    out IMember symbol
)
```

Parameters

fieldName	Type: System.String Name of the member.
symbol	Type: TwinCAT.TypeSystem.IMember [▶ 2065]. The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IMemberCollection.TryGetMember(String, IMember.) [▶ 2071]

Reference

MemberCollection Class [▶ 2281]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.108 MethodParamFlags Enumeration

Flag set specifying the MethodParameter context

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum MethodParamFlags
```

Members

	Member name	Value	Description
	In	1	Input Parameter (ADSMETHODPARAFLAG_IN)
	Out	2	Output Parameter (ADSMETHODPARAFLAG_OUT)
	ByReference	4	By reference Parameter (ADSMETHODPARAFLAG_BYREFERENCE)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.109 PrimitiveTypeFlags Enumeration

Enum PrimitiveTypeFlags

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum PrimitiveTypeFlags
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	System	1	System Type like Byte / Word / DWORD
	Unsigned	2	Primitive Type is Unsigned
	Bool	4	Boolean Value (maps to true and false)
	Float	8	Floating Point
	Date	16	Type represents a Date
	Time	32	Type represents a Time
	Numeric	64	Numeric value
	Bitset	128	Bitset
	MaskNumericUnsigned	66	Numeric / Unsigned Mask
	MaskDateTime	48	Date / Time Mask
	MaskSpecialType	67	Special type Mask

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.110 RawValueChangedEventArgs Class

Event args for the RawValueChanged [▶ 2254] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.TypeSystem.ValueChangedBaseEventArgs [▶ 2435]
      TwinCAT.TypeSystem.RawValueChangedEventArgs

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RawValueChangedEventArgs : ValueChangedBaseEventArgs
```

The RawValueChangedEventArgs type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Symbol [▶ 2438]	Gets the symbol. (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Value [▶ 2291]	New Value (byte[])

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.110.1 RawValueChangedEventArgs Properties

The RawValueChangedEventArgs [▶ 2289] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Symbol [▶ 2438]	Gets the symbol. (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Value [▶ 2291]	New Value (byte[])

Reference

RawValueChangedEventArgs Class [▶ 2289]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.110.1.1 RawValueChangedEventArgs.Value Property

New Value (byte[])

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemory Value { get; }
```

Property Value

Type: ReadOnlyMemory

Reference

RawValueChangedEventArgs Class [▶ 2289]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.110.2 RawValueChangedEventArgs Methods

The RawValueChangedEventArgs [▶ 2289] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

RawValueChangedEventArgs Class [▶ 2289]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.111 ReadOnlyDataTypeCollection Class

ReadOnly Collection of IDataType [▶ 1986] objects.

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IDataType [▶ 1986].
    TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection [▶ 2499].IDataType [▶ 1986].
      TwinCAT.TypeSystem.ReadOnlyDataTypeCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyDataTypeCollection : ReadOnlyDataTypeCollection<IDataType>,
    IDataTypeCollection, IDataTypeCollection<IDataType>, ICollection<IDataType>,
    IEnumerable<IDataType>, IEnumerable
```

The ReadOnlyDataTypeCollection type exposes the following members.

Constructors

	Name	Description
	ReadOnlyDataTypeCollection [▶ 2293]	Initializes a new instance of the ReadOnlyDataTypeCollection class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	Item.String. [▶ 2502]	Gets the element with the specified type name. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	ContainsType [▶ 2504]	Determines whether the specified name contains type. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2505]	Tries to get the Type with the specified name out of the collection. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.111.1 ReadOnlyDataTypeCollection Constructor

Initializes a new instance of the ReadOnlyDataTypeCollection [▶ 2291] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDataTypeCollection(
    DataTypeCollection<IDataType> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.Generic.DataTypeCollection [▶ 2442].IDataType [▶ 1986]. Collection of types.

Reference

ReadOnlyDataTypeCollection Class [▶ 2291]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.111.2 ReadOnlyDataTypeCollection Properties

The ReadOnlyDataTypeCollection [▶ 2291] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	Item.String. [▶ 2502]	Gets the element with the specified type name. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)

Reference

ReadOnlyDataTypeCollection Class [▶ 2291]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.111.3 ReadOnlyDataTypeCollection Methods

The ReadOnlyDataTypeCollection [▶ 2291] type exposes the following members.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	ContainsType [▶ 2504]	Determines whether the specified name contains type. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDataType [▶ 1986]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2505]	Tries to get the Type with the specified name out of the collection. (Inherited from ReadOnlyDataTypeCollection.T. [▶ 2499].)

Reference

ReadOnlyDataTypeCollection Class [▶ 2291]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112 ReadOnlyDimensionCollection Class

ReadOnly version of the DimensionCollection [▶ 1671]

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IDimension [▶ 1998].
    TwinCAT.TypeSystem.ReadOnlyDimensionCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyDimensionCollection : ReadOnlyCollection<IDimension>,
    IDimensionCollection, IList<IDimension>, ICollection<IDimension>,
    IEnumerable<IDimension>, IEnumerable
```

The ReadOnlyDimensionCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	ElementCount [▶ 2296]	Gets the Number of elements in all Dimensions
	Item	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	LowerBounds [▶ 2297]	Gets the lower bounds.
	UpperBounds [▶ 2297]	Gets the upper bounds.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	Empty [▶ 2299]	Returns an empty ReadOnlyDimensionCollection
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetDimensionLengths [▶ 2299]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.1 ReadOnlyDimensionCollection Properties

The ReadOnlyDimensionCollection [▶ 2295] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	ElementCount [▶ 2296]	Gets the Number of elements in all Dimensions
	Item	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	LowerBounds [▶ 2297]	Gets the lower bounds.
	UpperBounds [▶ 2297]	Gets the upper bounds.

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.1.1 ReadOnlyDimensionCollection.ElementCount Property

Gets the Number of elements in all Dimensions

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ElementCount { get; }
```

Property Value

Type: Int32

Implements

IDimensionCollection.ElementCount [▶ 2002]

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.1.2 ReadOnlyDimensionCollection.LowerBounds Property

Gets the lower bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] LowerBounds { get; }
```

Property Value

Type: .Int32.
The lower bounds.

Implements

IDimensionCollection.LowerBounds [▶ 2002]

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.1.3 ReadOnlyDimensionCollection.UpperBounds Property

Gets the upper bounds.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] UpperBounds { get; }
```

Property Value

Type: .Int32.
The upper bounds.

Implements

IDimensionCollection.UpperBounds [▶ 2002]

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.2 ReadOnlyDimensionCollection Methods

The ReadOnlyDimensionCollection [▶ 2295] type exposes the following members.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
S	Empty [▶ 2299]	Returns an empty ReadOnlyDimensionCollection [▶ 2295]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetDimensionLengths [▶ 2299]	Gets an array the specifies the Lengths of each Array Dimension
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IDimension [▶ 1998]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.2.1 ReadOnlyDimensionCollection.Empty Method

Returns an empty ReadOnlyDimensionCollection [▶ 2295]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ReadOnlyDimensionCollection Empty()
```

Return Value

Type: ReadOnlyDimensionCollection [▶ 2295]
ReadOnlyDimensionCollection.

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.112.2.2 ReadOnlyDimensionCollection.GetDimensionLengths Method

Gets an array the specifies the Lengths of each Array Dimension

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int[] GetDimensionLengths()
```

Return Value

Type: .Int32.
System.Int32[].

Implements

IDimensionCollection.GetDimensionLengths. [▶ 2004]

Reference

ReadOnlyDimensionCollection Class [▶ 2295]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113 ReadOnlyEnumValueCollection Class

Read only version of the EnumValueCollection.T. [▶ 1926]

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IEnumValue [▶ 2028].
    TwinCAT.TypeSystem.ReadOnlyEnumValueCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyEnumValueCollection : ReadOnlyCollection<IEnumValue>,
    IEnumValueCollection, IEnumValueCollection<IEnumValue, IConvertible>, ICollection<IEnumValue>,
    IEnumerable<IEnumValue>, IEnumerable
```

The ReadOnlyEnumValueCollection type exposes the following members.

Constructors

	Name	Description
	ReadOnlyEnumValueCollection [▶ 2301]	Initializes a new instance of the ReadOnlyEnumValueCollection.T. [▶ 2309] class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Item.String. [▶ 2302]	Gets or sets the element at the specified index.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)

Methods

	Name	Description
	Contains(String) [▶ 2305]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)

	Name	Description
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 2305]	Gets the Value Names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 2306]	Gets the values.
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 2306]	Parses the specified name.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryParse(String, IConvertible.) [▶ 2307]	Tries to pars the string value of the Enum.
	TryParse(String, IEnumValue.) [▶ 2308]	Tries to pars the string value of the Enum.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.1 ReadOnlyEnumValueCollection Constructor

Initializes a new instance of the ReadOnlyEnumValueCollection.T. [▶ 2309] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyEnumValueCollection(
    EnumValueCollection coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.EnumValueCollection [▶ 1908] The coll.

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.2 ReadOnlyEnumValueCollection Properties

The ReadOnlyEnumValueCollection [▶ 2300] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Item.String. [▶ 2302]	Gets or sets the element at the specified index.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.2.1 ReadOnlyEnumValueCollection.Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Item.String. [▶ 2302]	Gets or sets the element at the specified index.

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.Item Property (String)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name of the value

Return Value

Type: IConvertible
EnumValue<T>.

Implements

IEnumValueCollection.TEnumValue, TValue..Item.String. [▶ 2035]

Exceptions

Exception	Condition
NotImplementedException	
NotImplementedException	

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

Item Overload [▶ 2302]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3 ReadOnlyEnumValueCollection Methods

The ReadOnlyEnumValueCollection [▶ 2300] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2305]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)

	Name	Description
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 2305]	Gets the Value Names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 2306]	Gets the values.
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 2306]	Parses the specified name.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryParse(String, IConvertible.) [▶ 2307]	Tries to pars the string value of the Enum.
	TryParse(String, IEnumValue.) [▶ 2308]	Tries to pars the string value of the Enum.

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3.1 ReadOnlyEnumValueCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2305]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IEnumValue [▶ 2028]..)

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.Contains Method (String)

Determines whether [contains] [the specified name].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string value
)
```

Parameters

value	Type: System.String Value

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

IEnumValueCollection.TEnumValue, TValue..Contains(String) [▶ 2036]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

Contains Overload [▶ 2304]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3.2 ReadOnlyEnumValueCollection.GetNames Method

Gets the Value Names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetNames. [▶ 2037]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3.3 ReadOnlyEnumValueCollection.GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible[] GetValues()
```

Return Value

Type: .IConvertible.
T[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetValues. [▶ 2037]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3.4 ReadOnlyEnumValueCollection.Parse Method

Parses the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IConvertible Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: IConvertible
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Parse(String) [▶ 2038]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.113.3.5 ReadOnlyEnumValueCollection.TryParse Method

Overload List

	Name	Description
	TryParse(String, IConvertible.) [▶ 2307]	Tries to pars the string value of the Enum.
	TryParse(String, IEnumValue.) [▶ 2308]	Tries to pars the string value of the Enum.

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.TryParse Method (String, IConvertible.)

Tries to pars the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out IConvertible value
)
```

Parameters

strValue	Type: System.String The Value in string represention.
value	Type: System.IConvertible. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TryParse Overload [▶ 2307]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.TryParse Method (String, IEnumValue.)

Tries to pars the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out IEnumValue value
)
```

Parameters

strValue	Type: System.String The Value in string represention.
value	Type: TwinCAT.TypeSystem.IEnumValue [▶ 2028]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

ReadOnlyEnumValueCollection Class [▶ 2300]

TryParse Overload [▶ 2307]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114 ReadOnlyEnumValueCollection.T. Class

Read only version of the EnumValueCollection.T. [▶ 1926]

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.EnumValue [▶ 1901].T..
    TwinCAT.TypeSystem.ReadOnlyEnumValueCollection.T.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyEnumValueCollection<T> : ReadOnlyCollection<EnumValue<T>>,
    IEnumValueCollection<EnumValue<T>, T>, ICollection<EnumValue<T>>, IEnumerable<EnumValue<T>>,
    IEnumerable
where T : IConvertible
```

Type Parameters

T

The ReadOnlyEnumValueCollection.T. type exposes the following members.

Constructors

	Name	Description
	ReadOnlyEnumValueCollection.T. [▶ 2310]	Initializes a new instance of the ReadOnlyEnumValueCollection.T. class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	Item.String. [▶ 2312]	Gets the enumeration value T from its string representation.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)

Methods

	Name	Description
	Contains(String) [▶ 2313]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 2314]	Gets the Value Names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 2315]	Gets the values.
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T...)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 2315]	Parses the specified name.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryParse(String, EnumValue.T..) [▶ 2316]	Tries to parse the string value of the Enum.
	TryParse(String, T.) [▶ 2317]	Tries to parse the string value of the Enum.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.1 ReadOnlyEnumValueCollection.T. Constructor

Initializes a new instance of the ReadOnlyEnumValueCollection.T. [▶ 2309] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyEnumValueCollection(
    EnumValueCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.EnumValueCollection [▶ 1926].T [▶ 2309]. The coll.

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.2 ReadOnlyEnumValueCollection.T. Properties

The ReadOnlyEnumValueCollection.T. [▶ 2309] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	Item.String. [▶ 2312]	Gets the enumeration value T from its string representation.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.2.1 ReadOnlyEnumValueCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	Item.String. [▶ 2312]	Gets the enumeration value T from its string representation.

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.T..Item Property (String)

Gets the enumeration value T from its string representation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name of the enum value.

Return Value

Type: T [▶ 2309]
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Item.String. [▶ 2035]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

Item Overload [▶ 2311]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3 ReadOnlyEnumValueCollection.T. Methods

The ReadOnlyEnumValueCollection.T. [▶ 2309] generic type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2313]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetNames [▶ 2314]	Gets the Value Names.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	GetValues [▶ 2315]	Gets the values.
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Parse [▶ 2315]	Parses the specified name.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryParse(String, EnumValue.T..) [▶ 2316]	Tries to parse the string value of the Enum.
	TryParse(String, T.) [▶ 2317]	Tries to parse the string value of the Enum.

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3.1 ReadOnlyEnumValueCollection.T..Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2313]	Determines whether [contains] [the specified name].
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.EnumValue [▶ 1901].T [▶ 2309]...)

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.T..Contains Method (String)

Determines whether [contains] [the specified name].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string value
)
```

Parameters

value	Type: System.String Value

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

IEnumValueCollection.TEnumValue, TValue..Contains(String) [▶ 2036]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

Contains Overload [▶ 2313]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3.2 ReadOnlyEnumValueCollection.T..GetNames Method

Gets the Value Names.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string[] GetNames()
```

Return Value

Type: .String.
System.String[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetNames. [▶ 2037]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3.3 ReadOnlyEnumValueCollection.T..GetValues Method

Gets the values.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T[] GetValues()
```

Return Value

Type: .T [▶ 2309].
T[].

Implements

IEnumValueCollection.TEnumValue, TValue..GetValues. [▶ 2037]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3.4 ReadOnlyEnumValueCollection.T..Parse Method

Parses the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Parse(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 2309]
T.

Implements

IEnumValueCollection.TEnumValue, TValue..Parse(String) [▶ 2038]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.114.3.5 ReadOnlyEnumValueCollection.T..TryParse Method

Overload List

	Name	Description
	TryParse(String, EnumValue.T..) [▶ 2316]	Tries to parse the string value of the Enum.
	TryParse(String, T.) [▶ 2317]	Tries to parse the string value of the Enum.

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.T..TryParse Method (String, EnumValue.T..)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out EnumValue<T> value
)
```

Parameters

strValue	Type: System.String The Value in string represention.
value	Type: TwinCAT.TypeSystem.EnumValue [▶ 1901].T [▶ 2309].. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TryParse Overload [▶ 2316]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyEnumValueCollection.T..TryParse Method (String, T.)

Tries to parse the string value of the Enum.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryParse(
    string strValue,
    out T value
)
```

Parameters

strValue	Type: System.String The Value in string represention.
value	Type: T [▶ 2309]. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

IEnumValueCollection.TEnumValue, TValue..TryParse(String, TValue.) [▶ 2039]

Reference

ReadOnlyEnumValueCollection.T. Class [▶ 2309]

TryParse Overload [▶ 2316]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.115 ReadOnlyFieldCollection Class

Read only collection of IField [▶ 2040] objects

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IField [▶ 2040].
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [▶ 2505].IField [▶ 2040].

TwinCAT.TypeSystem.ReadOnlyFieldCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyFieldCollection : ReadOnlyInstanceCollection<IField>,
    IFieldCollection, IInstanceCollection<IField>, IList<IField>,
    ICollection<IField>, IEnumerable<IField>, IEnumerable
```

The ReadOnlyFieldCollection type exposes the following members.

Constructors

	Name	Description
	ReadOnlyFieldCollection [▶ 2319]	Initializes a new instance of the ReadOnlyMemberCollection [▶ 2322] class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetMember [▶ 2321]	Tries to get the specified member

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.115.1 ReadOnlyFieldCollection Constructor

Initializes a new instance of the ReadOnlyMemberCollection [▶ 2322] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyFieldCollection(
    FieldCollection members
)
```

Parameters

members	Type: TwinCAT.TypeSystem.FieldCollection [▶ 1945] The members.

Reference

ReadOnlyFieldCollection Class [▶ 2317]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.115.2 ReadOnlyFieldCollection Properties

The ReadOnlyFieldCollection [▶ 2317] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlyFieldCollection Class [▶ 2317]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.115.3 ReadOnlyFieldCollection Methods

The ReadOnlyFieldCollection [▶ 2317] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

	Name	Description
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IField [▶ 2040]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetMember [▶ 2321]	Tries to get the specified member

Reference

ReadOnlyFieldCollection Class [▶ 2317]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.115.3.1 ReadOnlyFieldCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMember(
    string fieldName,
    out IField symbol
)
```

Parameters

fieldName	Type: System.String Name of the member.
symbol	Type: TwinCAT.TypeSystem.IField [▶ 2040]. The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IFieldCollection.TryGetMember(String, IField.) [▶ 2045]

Reference

ReadOnlyFieldCollection Class [▶ 2317]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116 ReadOnlyMemberCollection Class

Read only collection of IMember [▶ 2065] objects

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IMember [▶ 2065].
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [▶ 2505].IMember [▶ 2065].
      TwinCAT.TypeSystem.ReadOnlyMemberCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyMemberCollection : ReadOnlyInstanceCollection<IMember>,
    IMemberCollection, IInstanceCollection<IMember>, IList<IMember>,
    ICollection<IMember>, IEnumerable<IMember>, IEnumerable
```

The ReadOnlyMemberCollection type exposes the following members.

Constructors

	Name	Description
	ReadOnlyMemberCollection [▶ 2323]	Initializes a new instance of the ReadOnlyMemberCollection class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
S	Empty [▶ 2326]	Returns an Empty ReadOnlyMemberCollection
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetMember [▶ 2326]	Tries to get the specified member

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116.1 ReadOnlyMemberCollection Constructor

Initializes a new instance of the ReadOnlyMemberCollection [▶ 2322] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMemberCollection(
    MemberCollection members
)
```

Parameters

members	Type: TwinCAT.TypeSystem.MemberCollection [▶ 2281] The members.

Reference

ReadOnlyMemberCollection Class [▶ 2322]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116.2 ReadOnlyMemberCollection Properties

The ReadOnlyMemberCollection [▶ 2322] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlyMemberCollection Class [▶ 2322]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116.3 ReadOnlyMemberCollection Methods

The ReadOnlyMemberCollection [▶ 2322] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
S	Empty [▶ 2326]	Returns an Empty ReadOnlyMemberCollection [▶ 2322]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IMember [▶ 2065]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetMember [▶ 2326]	Tries to get the specified member

Reference

ReadOnlyMemberCollection Class [▶ 2322]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116.3.1 ReadOnlyMemberCollection.Empty Method

Returns an Empty ReadOnlyMemberCollection [▶ 2322]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ReadOnlyMemberCollection Empty()
```

Return Value

Type: ReadOnlyMemberCollection [▶ 2322]
ReadOnlyMemberCollection.

Reference

ReadOnlyMemberCollection Class [▶ 2322]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.116.3.2 ReadOnlyMemberCollection.TryGetMember Method

Tries to get the specified member

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMember(
    string memberName,
    out IMember symbol
)
```

Parameters

memberName	Type: System.String Name of the member.
symbol	Type: TwinCAT.TypeSystem.IMember [▶ 2065]. The symbol.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IMemberCollection.TryGetMember(String, IMember.) [▶ 2071]

Reference

ReadOnlyMemberCollection Class [▶ 2322]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.117 ReadOnlyMethodParameterCollection Class

Read only RpcMethodParameterCollection [▶ 2384].

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IRpcMethodParameter [▶ 2133].
    TwinCAT.TypeSystem.ReadOnlyMethodParameterCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyMethodParameterCollection : ReadOnlyCollection<IRpcMethodParameter>,
    IRpcMethodParameterCollection, IList<IRpcMethodParameter>, ICollection<IRpcMethodParameter>,
    IEnumerable<IRpcMethodParameter>, IEnumerable
```

The ReadOnlyMethodParameterCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Item	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetLengthIsParameter [▶ 2329]	Gets the corresponding LengthIs parameter.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.117.1 ReadOnlyMethodParameterCollection Properties

The ReadOnlyMethodParameterCollection [▶ 2327] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Item	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)

Reference

ReadOnlyMethodParameterCollection Class [▶ 2327]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.117.2 ReadOnlyMethodParameterCollection Methods

The ReadOnlyMethodParameterCollection [▶ 2327] type exposes the following members.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetLengthIsParameter [▶ 2329]	Gets the corresponding LengthIs parameter.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethodParameter [▶ 2133]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ReadOnlyMethodParameterCollection Class [▶ 2327]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.117.2.1 ReadOnlyMethodParameterCollection.GetLengthIsParameter Method

Gets the corresponding LengthIs parameter.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```

Parameters

parameter	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The value parameter

Return Value

Type: IRpcMethodParameter [▶ 2133]
The LengthIs Parameter

Implements

IRpcMethodParameterCollection.GetLengthIsParameter(IRpcMethodParameter) [▶ 2139]

Reference

ReadOnlyMethodParameterCollection Class [▶ 2327]

TwinCAT.TypeSystem Namespace [▶ 1622]

IRpcMethodParameter.LengthIsParameterIndex [▶ 2135]

IRpcMethodParameter.HasLengthIsParameter [▶ 2135]

6.11.118 ReadOnlyRpcMethodCollection Class

Read only RpcMethodCollection [▶ 2364]

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.IRpcMethod [▶ 2123].
    TwinCAT.TypeSystem.ReadOnlyRpcMethodCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyRpcMethodCollection : ReadOnlyCollection<IRpcMethod>,
    IRpcMethodCollection, IList<IRpcMethod>, ICollection<IRpcMethod>,
    IEnumerable<IRpcMethod>, IEnumerable
```

The ReadOnlyRpcMethodCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2332]	Gets the IRpcMethod [▶ 2123] with the specified method name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)

Methods

	Name	Description
	Contains(String) [▶ 2334]	Determines whether this collection contains the specified method name.
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetMethod(Int32, IRpcMethod.) [▶ 2335]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2335]	Tries to get the specified method.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.118.1 ReadOnlyRpcMethodCollection Properties

The ReadOnlyRpcMethodCollection [▶ 2330] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2332]	Gets the IRpcMethod [▶ 2123] with the specified method name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.118.1.1 ReadOnlyRpcMethodCollection.Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Item.String. [▶ 2332]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyRpcMethodCollection.Item Property (String)

Gets the IRpcMethod [▶ 2123] with the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethod this[
    string methodName
] { get; }
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: IRpcMethod [▶ 2123]
RpcMethod.

Implements

IRpcMethodCollection.Item.String. [▶ 2129]

Exceptions

Exception	Condition
KeyNotFoundException	

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

Item Overload [▶ 2332]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.118.2 ReadOnlyRpcMethodCollection Methods

The ReadOnlyRpcMethodCollection [▶ 2330] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2334]	Determines whether this collection contains the specified method name.
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetMethod(Int32, IRpcMethod.) [▶ 2335]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2335]	Tries to get the specified method.

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.118.2.1 ReadOnlyRpcMethodCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2334]	Determines whether this collection contains the specified method name.

	Name	Description
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.IRpcMethod [▶ 2123]..)

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyRpcMethodCollection.Contains Method (String)

Determines whether this collection contains the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string methodName
)
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: Boolean
true if contained.; otherwise, false.

Implements

IRpcMethodCollection.Contains(String) [▶ 2131]

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

Contains Overload [▶ 2333]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.118.2.2 ReadOnlyRpcMethodCollection.TryGetMethod Method

Overload List

	Name	Description
	TryGetMethod(Int32, IRpcMethod.) [▶ 2335]	Tries to get the specified method.

	Name	Description
	TryGetMethod(String, IRpcMethod.) [▶ 2335]	Tries to get the specified method.

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyRpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

Parameters

vTableIndex	Type: System.Int32 vTableIndex.
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(Int32, IRpcMethod.) [▶ 2132]

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TryGetMethod Overload [▶ 2334]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyRpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMethod(
    string methodName,
    out IRpcMethod method
)
```

Parameters

methodName	Type: System.String Name of the method.
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(String, IRpcMethod.) [▶ 2133]

Reference

ReadOnlyRpcMethodCollection Class [▶ 2330]

TryGetMethod Overload [▶ 2334]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.119 ReadOnlySymbolCollection Class

ReadOnly collection containing ISymbol [▶ 2176] objects.

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.ISymbol [▶ 2176].
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [▶ 2505].ISymbol [▶ 2176].
      TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection [▶ 2522].ISymbol [▶ 2176].
        TwinCAT.TypeSystem.ReadOnlySymbolCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlySymbolCollection : ReadOnlySymbolCollection<ISymbol>,
    ISymbolCollection, ISymbolCollection<ISymbol>, IInstanceCollection<ISymbol>,
    IList<ISymbol>, ICollection<ISymbol>, IEnumerable<ISymbol>,
    IEnumerable
```

The ReadOnlySymbolCollection type exposes the following members.

Constructors

	Name	Description
	ReadOnlySymbolCollection [▶ 2338]	Initializes a new instance of the ReadOnlySymbolCollection class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)
S	Empty [▶ 2340]	Returns an Empty collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176].)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.119.1 ReadOnlySymbolCollection Constructor

Initializes a new instance of the ReadOnlySymbolCollection [▶ 2336] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlySymbolCollection(
    IInstanceCollection<ISymbol> symbols
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].ISymbol [▶ 2176]. The symbols.

Reference

ReadOnlySymbolCollection Class [▶ 2336]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.119.2 ReadOnlySymbolCollection Properties

The ReadOnlySymbolCollection [▶ 2336] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlySymbolCollection Class [▶ 2336]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.119.3 ReadOnlySymbolCollection Methods

The ReadOnlySymbolCollection [▶ 2336] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
S	Empty [▶ 2340]	Returns an Empty collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ISymbol [▶ 2176]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlySymbolCollection Class [▶ 2336]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.119.3.1 ReadOnlySymbolCollection.Empty Method

Returns an Empty collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ReadOnlySymbolCollection Empty()
```

Return Value

Type: ReadOnlySymbolCollection [▶ 2336]
ReadOnlySymbolCollection.

Reference

ReadOnlySymbolCollection Class [▶ 2336]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120 ReadOnlyTypeAttributeCollection Class

Read only version of the TypeAttributeCollection [▶ 2418]

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.ITypeAttribute [▶ 2209].
    TwinCAT.TypeSystem.ReadOnlyTypeAttributeCollection

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyTypeAttributeCollection : ReadOnlyCollection<ITypeAttribute>,
    ITypeAttributeCollection, IList<ITypeAttribute>, ICollection<ITypeAttribute>,
    IEnumerable<ITypeAttribute>, IEnumerable
```

The ReadOnlyTypeAttributeCollection type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2342]	Gets the String with the specified name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)

Methods

	Name	Description
	Contains(String) [▶ 2344]	Determines whether this ReadOnlyTypeAttributeCollection contains the specified attribute.
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
S	Empty [▶ 2345]	Returns an empty ReadOnlyTypeAttributeCollection
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

	Name	Description
	TryGetAttribute [▶ 2345]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2346]	Tries to get the specified Attribute value.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.1 ReadOnlyTypeAttributeCollection Properties

The ReadOnlyTypeAttributeCollection [▶ 2340] type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2342]	Gets the String with the specified name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.1.1 ReadOnlyTypeAttributeCollection.Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	Item.String. [▶ 2342]	Gets the String with the specified name.

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyTypeAttributeCollection.Item Property (String)

Gets the String with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: String
System.String.

Implements

ITypeAttributeCollection.Item.String. [▶ 2213]

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

Item Overload [▶ 2342]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.2 ReadOnlyTypeAttributeCollection Methods

The ReadOnlyTypeAttributeCollection [▶ 2340] type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2344]	Determines whether this ReadOnlyTypeAttributeCollection [▶ 2340] contains the specified attribute.
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
S	Empty [▶ 2345]	Returns an empty ReadOnlyTypeAttributeCollection [▶ 2340]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)

	Name	Description
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetAttribute [▶ 2345]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2346]	Tries to get the specified Attribute value.

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.2.1 ReadOnlyTypeAttributeCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2344]	Determines whether this ReadOnlyTypeAttributeCollection [▶ 2340] contains the specified attribute.
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.ITypeAttribute [▶ 2209]..)

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

ReadOnlyTypeAttributeCollection.Contains Method (String)

Determines whether this ReadOnlyTypeAttributeCollection [▶ 2340] contains the specified attribute.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

ITypeAttributeCollection.Contains(String) [▶ 2214]

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

Contains Overload [▶ 2344]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.2.2 ReadOnlyTypeAttributeCollection.Empty Method

Returns an empty ReadOnlyTypeAttributeCollection [▶ 2340]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ReadOnlyTypeAttributeCollection Empty()
```

Return Value

Type: ReadOnlyTypeAttributeCollection [▶ 2340]
ReadOnlyTypeAttributeCollection.

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.2.3 ReadOnlyTypeAttributeCollection.TryGetAttribute Method

Tries to get the specified ITypeAttribute [▶ 2209]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetAttribute(
    string name,
    out ITypeAttribute attribute
)
```

Parameters

name	Type: System.String The name of the ITypeAttribute [▶ 2209].
attribute	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209]. The attribute.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

ITypeAttributeCollection.TryGetAttribute(String, ITypeAttribute.) [▶ 2215]

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.120.2.4 ReadOnlyTypeAttributeCollection.TryGetValue Method

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetValue(
    string name,
    out string value
)
```

Parameters

name	Type: System.String The name.
value	Type: System.String. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

ITypeAttributeCollection.TryGetValue(String, String.) [▶ 2215]

Reference

ReadOnlyTypeAttributeCollection Class [▶ 2340]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.121 ResultDataTypes Class

Class representing the asynchronous result of reading a IDataTypeCollection [▶ 1993] via ADS. Implements the ResultValue.TValue. [▶ 1029]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultValue [▶ 1029].IDataTypeCollection [▶ 1995].IDataType [▶ 1986]..
      TwinCAT.TypeSystem.ResultDataTypes

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultDataTypes : ResultValue<IDataTypeCollection<IDataType>>
```

The ResultDataTypes type exposes the following members.

Constructors

	Name	Description
	ResultDataTypes [▶ 2348]	Initializes a new instance of the ResultDataTypes class.

Properties

	Name	Description
	DataTypes [▶ 2349]	Gets the data types.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

The Result value of this operation can be accessed by the DataTypes [▶ 2349] property.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.Ads.ResultValue.TValue. [▶ 1029]

6.11.121.1 ResultDataTypes Constructor

Initializes a new instance of the ResultDataTypes [▶ 2347] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultDataTypes(
    AdsErrorCode errorCode,
    IDataTypeCollection<IDataType> dataTypes
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
dataTypes	Type: TwinCAT.TypeSystem.IDataTypeCollection [▶ 1995].IDataType [▶ 1986]. The datatype collection..

Reference

ResultDataTypes Class [▶ 2347]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.121.2 ResultDataTypes Properties

The ResultDataTypes [▶ 2347] type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2349]	Gets the data types.
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Reference

ResultDataTypes Class [▶ 2347]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.121.2.1 ResultDataTypes.DataTypes Property

Gets the data types.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataTypeCollection<IDataType> DataTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].IDataType [▶ 1986].
The data types.

Reference

ResultDataTypes Class [▶ 2347]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.121.3 ResultDataTypes Methods

The ResultDataTypes [▶ 2347] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultDataTypes Class [▶ 2347]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.122 ResultDynamicSymbols Class

Class representing the asynchronous result of reading a dynamic symbol collection via ADS. Implements the ResultValue.TValue. [▶ 1029]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultValue [▶ 1029].IDynamicSymbolsCollection [▶ 2010].
      TwinCAT.TypeSystem.ResultSymbols [▶ 2356].IDynamicSymbolsCollection [▶ 2010].
        TwinCAT.TypeSystem.ResultDynamicSymbols

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultDynamicSymbols : ResultSymbols<IDynamicSymbolsCollection>
```

The ResultDynamicSymbols type exposes the following members.

Constructors

	Name	Description
	ResultDynamicSymbols [▶ 2351]	Initializes a new instance of the ResultDynamicSymbols class.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)

	Name	Description
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T. [▶ 2356].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This result class is used to return Symbol instances of dynamic nature in an IDynamicSymbolsCollection [▶ 2010]. The value/result of the completed operation can be returned by the Symbols [▶ 2358] property.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.Ads.ResultValue.TValue. [▶ 1029]

6.11.122.1 ResultDynamicSymbols Constructor

Initializes a new instance of the ResultDynamicSymbols [▶ 2350] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultDynamicSymbols(
    AdsErrorCode errorCode,
    IDynamicSymbolsCollection symbols
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
symbols	Type: TwinCAT.TypeSystem.IDynamicSymbolsCollection [▶ 2010] The result symbols.

Reference

ResultDynamicSymbols Class [▶ 2350]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.122.2 ResultDynamicSymbols Properties

The ResultDynamicSymbols [▶ 2350] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T. [▶ 2356].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Reference

ResultDynamicSymbols Class [▶ 2350]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.122.3 ResultDynamicSymbols Methods

The ResultDynamicSymbols [▶ 2350] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultDynamicSymbols Class [▶ 2350]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.123 ResultSymbols Class

Class representing the asynchronous result of reading an symbol collection of type ISymbolCollection.T. [▶ 2185] via ADS. Implements the ResultValue.TValue. [▶ 1029]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultValue [▶ 1029].ISymbolCollection [▶ 2185].ISymbol [▶ 2176]..
      TwinCAT.TypeSystem.ResultSymbols [▶ 2356].ISymbolCollection [▶ 2185].ISymbol [▶ 2176]..
        TwinCAT.TypeSystem.ResultSymbols

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSymbols : ResultSymbols<ISymbolCollection<ISymbol>>
```

The ResultSymbols type exposes the following members.

Constructors

	Name	Description
	ResultSymbols [▶ 2354]	Initializes a new instance of the ResultSymbols class.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T. [▶ 2356].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This result class is used to return generic Symbol instances. in an enumerable class. The value/result of the completed operation can be returned by the Symbols [▶ 2358] property.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.Ads.ResultValue.TValue. [▶ 1029]

6.11.123.1 ResultSymbols Constructor

Initializes a new instance of the ResultSymbols [▶ 2353] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultSymbols(
    AdsErrorCode errorCode,
    ISymbolCollection<ISymbol> symbols
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
symbols	Type: TwinCAT.TypeSystem.ISymbolCollection [▶ 2185].ISymbol [▶ 2176]. The result symbols.

Reference

ResultSymbols Class [▶ 2353]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.123.2 ResultSymbols Properties

The ResultSymbols [▶ 2353] type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation. (Inherited from ResultSymbols.T. [▶ 2356].)
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Reference

ResultSymbols Class [▶ 2353]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.123.3 ResultSymbols Methods

The ResultSymbols [▶ 2353] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSymbols Class [▶ 2353]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.124 ResultSymbols.T. Class

Class representing the asynchronous result of reading a symbol enumeration of type IEnumerable.T. via ADS. Implements the ResultValue.TValue. [▶ 1029]

Inheritance Hierarchy

System.Object
  TwinCAT.Ads.ResultAds [▶ 989]
    TwinCAT.Ads.ResultValue [▶ 1029].T.
      TwinCAT.TypeSystem.ResultSymbols.T.
        TwinCAT.TypeSystem.ResultDynamicSymbols [▶ 2350]
        TwinCAT.TypeSystem.ResultSymbols [▶ 2353]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultSymbols<T> : ResultValue<T>
where T : class, Object, IEnumerable<ISymbol>
```

Type Parameters

T	The type parameter is an IEnumerable.T..

The ResultSymbols.T. type exposes the following members.

Constructors

	Name	Description
	ResultSymbols.T. [▶ 2357]	Initializes a new instance of the ResultSymbols.T. class.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation.
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This result class is used to return generic Symbol instances. in an enumerable class. The value/result of the completed operation can be returned by the Symbols [▶ 2358] property.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.Ads.ResultValue.TValue. [▶ 1029]

6.11.124.1 ResultSymbols.T. Constructor

Initializes a new instance of the ResultSymbols.T. [▶ 2356] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultSymbols(
    AdsErrorCode errorCode,
    T symbols
)
```

Parameters

errorCode	Type: TwinCAT.Ads.AdsErrorCode [▶ 575] The error code.
symbols	Type: T [▶ 2356] The symbols.

Reference

ResultSymbols.T. Class [▶ 2356]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.124.2 ResultSymbols.T. Properties

The ResultSymbols.T. [▶ 2356] generic type exposes the following members.

Properties

	Name	Description
	ErrorCode [▶ 992]	Gets the ADS Error code bound to this Result [▶ 989] object. (Inherited from ResultAds [▶ 989].)
	Failed [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is failed. (Inherited from ResultAds [▶ 989].)
	Succeeded [▶ 993]	Gets a value indicating whether the ResultAds [▶ 989] state is succeeded. (Inherited from ResultAds [▶ 989].)
	Symbols [▶ 2358]	Get the Symbols enumerable (T) as result of an asynchronous operation.
	Value [▶ 1032]	The value object. (Inherited from ResultValue.TValue. [▶ 1029].)

Reference

ResultSymbols.T. Class [▶ 2356]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.124.2.1 ResultSymbols.T..Symbols Property

Get the Symbols enumerable (T) as result of an asynchronous operation.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Symbols { get; }
```

Property Value

Type: T [▶ 2356]
The symbols.

Reference

ResultSymbols.T. Class [▶ 2356]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.124.3 ResultSymbols.T. Methods

The ResultSymbols.T. [▶ 2356] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 997]	Sets the error state of this ResultAds [▶ 989] (Inherited from ResultAds [▶ 989].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultSymbols.T. Class [▶ 2356]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125 RpcInvokeException Class

Class RpcInvokeException. Implements the SymbolException [▶ 2401]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]
        TwinCAT.TypeSystem.RpcInvokeException

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class RpcInvokeException : SymbolException
```

The RpcInvokeException type exposes the following members.

Constructors

	Name	Description
	RpcInvokeException (SerializationInfo, StreamingContext) [▶ 2361]	Initializes a new instance of the RpcInvokeException class.
	RpcInvokeException (IRpcStructInstance, String, Int32) [▶ 2362]	Initializes a new instance of the RpcInvokeException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2363]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides SymbolException.GetObjectData(SerializationInfo, StreamingContext) [▶ 2412].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

TwinCAT.TypeSystem.SymbolException [▶ 2401]

6.11.125.1 RpcInvokeException Constructor

Overload List

	Name	Description
	RpcInvokeException (SerializationInfo, StreamingContext) [▶ 2361]	Initializes a new instance of the RpcInvokeException [▶ 2359] class.
	RpcInvokeException (IRpcStructInstance, String, Int32) [▶ 2362]	Initializes a new instance of the RpcInvokeException [▶ 2359] class.

Reference

RpcInvokeException Class [▶ 2359]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.1.1 RpcInvokeException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the RpcInvokeException [▶ 2359] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected RpcInvokeException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

RpcInvokeException Class [▶ 2359]

RpcInvokeException Overload [▶ 2361]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.1.2 RpcInvokeException Constructor (IRpcStructInstance, String, Int32)

Initializes a new instance of the RpcInvokeException [▶ 2359] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RpcInvokeException(
    IRpcStructInstance rpcInstance,
    string methodName,
    int errorCode
)
```

Parameters

rpcInstance	Type: TwinCAT.TypeSystem.IRpcStructInstance [▶ 2140] The RPC instance.
methodName	Type: System.String Name of the method.
errorCode	Type: System.Int32 The error code.

Reference

RpcInvokeException Class [▶ 2359]

RpcInvokeException Overload [▶ 2361]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.2 RpcInvokeException Properties

The RpcInvokeException [▶ 2359] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)

	Name	Description
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

RpcInvokeException Class [▶ 2359]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.3 RpcInvokeException Methods

The RpcInvokeException [▶ 2359] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2363]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides SymbolException.GetObjectData(SerializationInfo, StreamingContext) [▶ 2412].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

RpcInvokeException Class [▶ 2359]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.3.1 RpcInvokeException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

RpcInvokeException Class [▶ 2359]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.125.4 RpcInvokeException Events

The RpcInvokeException [▶ 2359] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

RpcInvokeException Class [▶ 2359]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126 RpcMethodCollection Class

Collection of RpcMethods. [▶ 2123]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.RpcMethodCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RpcMethodCollection : IRpcMethodCollection,
    IList<IRpcMethod>, ICollection<IRpcMethod>, IEnumerable<IRpcMethod>,
    IEnumerable
```

The RpcMethodCollection type exposes the following members.

Properties

	Name	Description
	Count [▶ 2366]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2367]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2368]	Gets or sets the element at the specified index.
	Item.String. [▶ 2368]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Methods

	Name	Description
	Add [▶ 2370]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 2371]	Gets a read only collection of this RpcMethodCollection
	Clear [▶ 2371]	Removes all items from the ICollection.T..
	Contains(String) [▶ 2372]	Determines whether this collection contains the specified method name.
	Contains(IRpcMethod) [▶ 2373]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2373]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2374]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2374]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2375]	Inserts an item to the IList.T. at the specified index.

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2376]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2376]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetMethod(Int32, IRpcMethod.) [▶ 2377]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2378]	Tries to get the specified method.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.1 RpcMethodCollection Properties

The RpcMethodCollection [▶ 2364] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2366]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2367]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2368]	Gets or sets the element at the specified index.
	Item.String. [▶ 2368]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.1.1 RpcMethodCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.1.2 RpcMethodCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.1.3 RpcMethodCollection.Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 2368]	Gets or sets the element at the specified index.
	Item.String. [▶ 2368]	Gets the IRpcMethod [▶ 2123] with the specified method name.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethod this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: IRpcMethod [▶ 2123]
RpcMethod.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

RpcMethodCollection Class [▶ 2364]

Item Overload [▶ 2367]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.Item Property (String)

Gets the IRpcMethod [▶ 2123] with the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethod this[
    string methodName
] { get; }
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: IRpcMethod [▶ 2123]
RpcMethod.

Implements

IRpcMethodCollection.Item.String. [▶ 2129]

Exceptions

Exception	Condition
KeyNotFoundException	

Reference

RpcMethodCollection Class [▶ 2364]

Item Overload [▶ 2367]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2 RpcMethodCollection Methods

The RpcMethodCollection [▶ 2364] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2370]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 2371]	Gets a read only collection of this RpcMethodCollection [▶ 2364]
	Clear [▶ 2371]	Removes all items from the ICollection.T..
	Contains(String) [▶ 2372]	Determines whether this collection contains the specified method name.
	Contains(IRpcMethod) [▶ 2373]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2373]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2374]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2374]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2375]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2376]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2376]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetMethod(Int32, IRpcMethod.) [▶ 2377]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2378]	Tries to get the specified method.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.1 RpcMethodCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    IRpcMethod item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.2 RpcMethodCollection.AsReadOnly Method

Gets a read only collection of this RpcMethodCollection [▶ 2364]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyRpcMethodCollection AsReadOnly()
```

Field Value

Type: ReadOnlyRpcMethodCollection [▶ 2330]
Returns a read only version of this RpcMethodCollection [▶ 2364]

Return Value

Type: ReadOnlyRpcMethodCollection [▶ 2330]
ReadOnlyRpcMethodCollection.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.3 RpcMethodCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.4 RpcMethodCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2372]	Determines whether this collection contains the specified method name.
	Contains(IRpcMethod) [▶ 2373]	Determines whether the ICollection.T. contains a specific value.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.Contains Method (String)

Determines whether this collection contains the specified method name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string methodName
)
```

Parameters

methodName	Type: System.String Name of the method.

Return Value

Type: Boolean
true if contained.; otherwise, false.

Implements

IRpcMethodCollection.Contains(String) [▶ 2131]

Reference

RpcMethodCollection Class [▶ 2364]

Contains Overload [▶ 2372]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.Contains Method (IRpcMethod)

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    IRpcMethod item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

RpcMethodCollection Class [▶ 2364]

Contains Overload [▶ 2372]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.5 RpcMethodCollection.CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    IRpcMethod[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.6 RpcMethodCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<IRpcMethod> GetEnumerator()
```

Return Value

Type: IEnumerator.IRpcMethod [▶ 2123].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.7 RpcMethodCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    IRpcMethod item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.8 RpcMethodCollection.Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    IRpcMethod item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.9 RpcMethodCollection.Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    IRpcMethod item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123] The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.1 RpcMethodCollection.RemoveAt Method 0

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.126.2.1 RpcMethodCollection.TryGetMethod Method 1

Overload List

	Name	Description
	TryGetMethod(Int32, IRpcMethod.) [▶ 2377]	Tries to get the specified method.
	TryGetMethod(String, IRpcMethod.) [▶ 2378]	Tries to get the specified method.

Reference

RpcMethodCollection Class [▶ 2364]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.TryGetMethod Method (Int32, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMethod(
    int vTableIndex,
    out IRpcMethod method
)
```

Parameters

vTableIndex	Type: System.Int32 VTable index.

method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(Int32, IRpcMethod.) [▶ 2132]

Reference

RpcMethodCollection Class [▶ 2364]

TryGetMethod Overload [▶ 2377]

TwinCAT.TypeSystem Namespace [▶ 1622]

RpcMethodCollection.TryGetMethod Method (String, IRpcMethod.)

Tries to get the specified method.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetMethod(
    string methodName,
    out IRpcMethod method
)
```

Parameters

methodName	Type: System.String Name of the method.
method	Type: TwinCAT.TypeSystem.IRpcMethod [▶ 2123]. The method if fund, NULL otherwise.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

IRpcMethodCollection.TryGetMethod(String, IRpcMethod.) [▶ 2133]

Reference

RpcMethodCollection Class [▶ 2364]

TryGetMethod Overload [▶ 2377]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127 RpcMethodNotSupportedException Class

Symbol Exception

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]
        TwinCAT.TypeSystem.RpcMethodNotSupportedException

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class RpcMethodNotSupportedException : SymbolException
```

The RpcMethodNotSupportedException type exposes the following members.

Constructors

	Name	Description
	RpcMethodNotSupportedException(Int32, ISymbol) [▶ 2381]	Initializes a new instance of the RpcMethodNotSupportedException class.
	RpcMethodNotSupportedException(SerializationInfo, StreamingContext) [▶ 2382]	Initializes a new instance of the RpcMethodNotSupportedException class.
	RpcMethodNotSupportedException(String, ISymbol) [▶ 2382]	Initializes a new instance of the RpcMethodNotSupportedException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.1 RpcMethodNotSupportedException Constructor

Overload List

	Name	Description
	RpcMethodNotSupportedException(Int32, ISymbol) [▶ 2381]	Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.
	RpcMethodNotSupportedException(SerializationInfo, StreamingContext) [▶ 2382]	Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.
	RpcMethodNotSupportedException(String, ISymbol) [▶ 2382]	Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.

Reference

RpcMethodNotSupportedException Class [▶ 2379]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.1.1 RpcMethodNotSupportedException Constructor (Int32, ISymbol)

Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RpcMethodNotSupportedException(
    int vTableIndex,
    ISymbol symbol
)
```

Parameters

vTableIndex	Type: System.Int32 Index of the v table.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

RpcMethodNotSupportedException Class [▶ 2379]

RpcMethodNotSupportedException Overload [▶ 2381]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.1.2 RpcMethodNotSupportedException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected RpcMethodNotSupportedException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Exceptions

Exception	Condition
NotImplementedException	

Reference

RpcMethodNotSupportedException Class [▶ 2379]

RpcMethodNotSupportedException Overload [▶ 2381]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.1.3 RpcMethodNotSupportedException Constructor (String, ISymbol)

Initializes a new instance of the RpcMethodNotSupportedException [▶ 2379] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public RpcMethodNotSupportedException(
    string methodName,
    ISymbol symbol
)
```

Parameters

methodName	Type: System.String Name of the method.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

RpcMethodNotSupportedException Class [▶ 2379]

RpcMethodNotSupportedException Overload [▶ 2381]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.2 RpcMethodNotSupportedException Properties

The RpcMethodNotSupportedException [▶ 2379] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

RpcMethodNotSupportedException Class [▶ 2379]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.3 RpcMethodNotSupportedException Methods

The RpcMethodNotSupportedException [▶ 2379] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

RpcMethodNotSupportedException Class [▶ 2379]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.127.4 RpcMethodNotSupportedException Events

The RpcMethodNotSupportedException [▶ 2379] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

RpcMethodNotSupportedException Class [▶ 2379]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128 RpcMethodParameterCollection Class

Collection of RPC method parameters

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.RpcMethodParameterCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class RpcMethodParameterCollection : IRpcMethodParameterCollection,
    IList<IRpcMethodParameter>, ICollection<IRpcMethodParameter>, IEnumerable<IRpcMethodParameter>,
    IEnumerable
```

The RpcMethodParameterCollection type exposes the following members.

Properties

	Name	Description
	Count [▶ 2386]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2386]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 2387]	Gets or sets the element at the specified index.

Methods

	Name	Description
	Add [▶ 2388]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 2389]	Returns a read only version of this RpcMethodParameterCollection
	Clear [▶ 2390]	Removes all items from the ICollection.T..
	Contains [▶ 2390]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2391]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2391]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetLengthIsParameter [▶ 2392]	Gets the length is parameter.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2392]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2393]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2394]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2394]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.1 RpcMethodParameterCollection Properties

The RpcMethodParameterCollection [▶ 2384] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2386]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2386]	Gets a value indicating whether the ICollection.T. is read-only.
	Item [▶ 2387]	Gets or sets the element at the specified index.

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.1.1 RpcMethodParameterCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.1.2 RpcMethodParameterCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.1.3 RpcMethodParameterCollection.Item Property

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameter this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: IRpcMethodParameter [▶ 2133]
RpcMethodParameter.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2 RpcMethodParameterCollection Methods

The RpcMethodParameterCollection [▶ 2384] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2388]	Adds an item to the ICollection.T..
	AsReadOnly [▶ 2389]	Returns a read only version of this RpcMethodParameterCollection [▶ 2384]
	Clear [▶ 2390]	Removes all items from the ICollection.T..
	Contains [▶ 2390]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2391]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2391]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetLengthIsParameter [▶ 2392]	Gets the length is parameter.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2392]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2393]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2394]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2394]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.1 RpcMethodParameterCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    IRpcMethodParameter item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.2 RpcMethodParameterCollection.AsReadOnly Method

Returns a read only version of this RpcMethodParameterCollection [▶ 2384]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyMethodParameterCollection AsReadOnly()
```

Field Value

Type: ReadOnlyMethodParameterCollection [▶ 2327]
Collection as read only version.

Return Value

Type: ReadOnlyMethodParameterCollection [▶ 2327]
ReadOnlyMethodParameterCollection.

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.3 RpcMethodParameterCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.4 RpcMethodParameterCollection.Contains Method

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    IRpcMethodParameter item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.5 RpcMethodParameterCollection.CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    IRpcMethodParameter[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.6 RpcMethodParameterCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<IRpcMethodParameter> GetEnumerator()
```

Return Value

Type: IEnumerator.IRpcMethodParameter [▶ 2133].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.7 RpcMethodParameterCollection.GetLengthIsParameter Method

Gets the length is parameter.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IRpcMethodParameter GetLengthIsParameter(
    IRpcMethodParameter parameter
)
```

Parameters

parameter	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The parameter.

Return Value

Type: IRpcMethodParameter [▶ 2133]
IRpcMethodParameter.

Implements

IRpcMethodParameterCollection.GetLengthIsParameter(IRpcMethodParameter) [▶ 2139]

Exceptions

Exception	Condition
ArgumentNullException	parameter
ArgumentException	Parameter is not contained in ParameterList - parameter

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.8 RpcMethodParameterCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    IRpcMethodParameter item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.9 RpcMethodParameterCollection.Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    IRpcMethodParameter item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.1 RpcMethodParameterCollection.Remove Method 0

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    IRpcMethodParameter item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.IRpcMethodParameter [▶ 2133] The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.128.2.1 RpcMethodParameterCollection.RemoveAt Method 1

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

RpcMethodParameterCollection Class [▶ 2384]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.129 StringConvertMode Enumeration

Enum StringConvertMode

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public enum StringConvertMode
```

Members

	Member name	Value	Description
	FixedLength	0	Fixed Length String
	FixedLengthZeroTerminated	1	Fixed Length String that can be terminated with '\0'
	ZeroTerminated	2	'\0' terminated dynamic length string
	LengthPrefix	3	Length Prefix (number of following bytes as uint 4-Byte)

Remarks

The StringConvertMode is used to specify, how Strings will be marshalled / demarshalled.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

StringMarshaler

6.11.130 SymbolAccessRights Enumeration

Enum specifying Access Rights to symbols

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum SymbolAccessRights
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Read	1	Read-Access
	Write	2	Write-Access
	MethodInvoke	4	Right to Invoke Methods / RPC Invoke
	ReadWrite	3	Read / Write Access
	All	7	Full Access

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131 SymbolCollection Class

Interface represents a collection of ISymbol [▶ 2176] objects.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection [▶ 2460].ISymbol [▶ 2176].
    TwinCAT.TypeSystem.Generic.SymbolCollection [▶ 2526].ISymbol [▶ 2176].
      TwinCAT.TypeSystem.SymbolCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SymbolCollection : SymbolCollection<ISymbol>,
    ISymbolCollection, ISymbolCollection<ISymbol>, IInstanceCollection<ISymbol>,
    IList<ISymbol>, ICollection<ISymbol>, IEnumerable<ISymbol>,
    IEnumerable
```

The SymbolCollection type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the LIst of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2400]	Returns a Read only version of this collection (shallow copy).
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2400]	Clones this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2401]	Returns an Empty Collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstances [▶ 2531]	Try to get instances with predicate function (Inherited from SymbolCollection.T. [▶ 2526].)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131.1 SymbolCollection Properties

The SymbolCollection [▶ 2396] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the LIst of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Reference

SymbolCollection Class [▶ 2396]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131.2 SymbolCollection Methods

The SymbolCollection [▶ 2396] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2400]	Returns a Read only version of this collection (shallow copy).
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2400]	Clones this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2401]	Returns an Empty Collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstances [▶ 2531]	Try to get instances with predicate function (Inherited from SymbolCollection.T. [▶ 2526].)

Reference

SymbolCollection Class [▶ 2396]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131.2.1 SymbolCollection.AsReadOnly Method

Returns a Read only version of this collection (shallow copy).

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlySymbolCollection AsReadOnly()
```

Return Value

Type: ReadOnlySymbolCollection [▶ 2336]
Read only collection.

Reference

SymbolCollection Class [▶ 2396]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131.2.2 SymbolCollection.Clone Method

Clones this instance.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolCollection Clone()
```

Return Value

Type: SymbolCollection [▶ 2396]
Cloned SymbolCollection [▶ 2396].

Reference

SymbolCollection Class [▶ 2396]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.131.2.3 SymbolCollection.Empty Method

Returns an Empty Collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SymbolCollection Empty()
```

Return Value

Type: SymbolCollection [▶ 2396]
SymbolCollection.

Reference

SymbolCollection Class [▶ 2396]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132 SymbolException Class

Symbol bound exceptions

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException
        TwinCAT.TypeSystem.CannotAccessVirtualSymbolException [▶ 1641]
        TwinCAT.TypeSystem.InsufficientAccessRightsException [▶ 2076]
        TwinCAT.TypeSystem.RpcInvokeException [▶ 2359]
        TwinCAT.TypeSystem.RpcMethodNotSupportedException [▶ 2379]
        TwinCAT.ValueAccess.CannotAccessValueException [▶ 2541]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public class SymbolException : AdsException
```

The SymbolException type exposes the following members.

Constructors

	Name	Description
	SymbolException(ISymbol) [▶ 2404]	Initializes a new instance of the SymbolException class.
	SymbolException(SerializationInfo, StreamingContext) [▶ 2405]	Initializes a new instance of the SymbolException class.
	SymbolException(String, ISymbol) [▶ 2405]	Initializes a new instance of the SymbolException class.
	SymbolException(ISymbol, Exception) [▶ 2406]	Initializes a new instance of the SymbolException class.
	SymbolException(ISymbol, Int32) [▶ 2407]	Initializes a new instance of the SymbolException class.
	SymbolException(String, ISymbol, Exception) [▶ 2407]	Initializes a new instance of the SymbolException class.
	SymbolException(String, ISymbol, Int32) [▶ 2408]	Initializes a new instance of the SymbolException class.
	SymbolException(ISymbol, Int32, Exception) [▶ 2408]	Initializes a new instance of the SymbolException class.
	SymbolException(String, ISymbol, Int32, Exception) [▶ 2409]	Initializes a new instance of the SymbolException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)

	Name	Description
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path.
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol.
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1 SymbolException Constructor

Overload List

	Name	Description
	SymbolException(ISymbol) [▶ 2404]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(SerializationInfo, StreamingContext) [▶ 2405]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(String, ISymbol) [▶ 2405]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(ISymbol, Exception) [▶ 2406]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(ISymbol, Int32) [▶ 2407]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(String, ISymbol, Exception) [▶ 2407]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(String, ISymbol, Int32) [▶ 2408]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(ISymbol, Int32, Exception) [▶ 2408]	Initializes a new instance of the SymbolException [▶ 2401] class.
	SymbolException(String, ISymbol, Int32, Exception) [▶ 2409]	Initializes a new instance of the SymbolException [▶ 2401] class.

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.1 SymbolException Constructor (ISymbol)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.2 SymbolException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected SymbolException(
    SerializationInfo serializationInfo,
    StreamingContext streamingContext
)
```

Parameters

serializationInfo	Type: System.Runtime.Serialization.SerializationInfo The serialization information.
streamingContext	Type: System.Runtime.Serialization.StreamingContext The streaming context.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.3 SymbolException Constructor (String, ISymbol)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    string message,
    ISymbol symbol
)
```

Parameters

message	Type: System.String The message.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.4 SymbolException Constructor (ISymbol, Exception)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    ISymbol symbol,
    Exception innerException
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
innerException	Type: System.Exception The inner exception.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.5 SymbolException Constructor (ISymbol, Int32)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    ISymbol symbol,
    int errorCode
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
errorCode	Type: System.Int32 The error code.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.6 SymbolException Constructor (String, ISymbol, Exception)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    string message,
    ISymbol symbol,
    Exception innerException
)
```

Parameters

message	Type: System.String The message.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
innerException	Type: System.Exception The inner exception.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.7 SymbolException Constructor (String, ISymbol, Int32)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    string message,
    ISymbol symbol,
    int errorCode
)
```

Parameters

message	Type: System.String The message.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
errorCode	Type: System.Int32 The error code.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.8 SymbolException Constructor (ISymbol, Int32, Exception)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    ISymbol symbol,
    int errorCode,
    Exception innerException
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
errorCode	Type: System.Int32 The error code.
innerException	Type: System.Exception The inner exception.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.1.9 SymbolException Constructor (String, ISymbol, Int32, Exception)

Initializes a new instance of the SymbolException [▶ 2401] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolException(
    string message,
    ISymbol symbol,
    int errorCode,
    Exception innerException
)
```

Parameters

message	Type: System.String The message.
symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
errorCode	Type: System.Int32 The error code.
innerException	Type: System.Exception The inner exception.

Reference

SymbolException Class [▶ 2401]

SymbolException Overload [▶ 2404]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.2 SymbolException Properties

The SymbolException [▶ 2401] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path.
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol.
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.2.1 SymbolException.InstancePath Property

Gets the instance path.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string InstancePath { get; }
```

Property Value

Type: String
The instance path.

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.2.2 SymbolException.Symbol Property

Gets the symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Symbol { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The symbol.

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.3 SymbolException Methods

The SymbolException [▶ 2401] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext).)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.3.1 SymbolException.GetObjectData Method

When overridden in a derived class, sets the SerializationInfo with information about the exception.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

Parameters

info	Type: System.Runtime.Serialization.SerializationInfo The SerializationInfo that holds the serialized object data about the exception being thrown.
context	Type: System.Runtime.Serialization.StreamingContext The StreamingContext that contains contextual information about the source or destination.

Implements

ISerializable.GetObjectData(SerializationInfo, StreamingContext)
_Exception.GetObjectData(SerializationInfo, StreamingContext)

Exceptions

Exception	Condition
ArgumentNullException	info

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.132.4 SymbolException Events

The SymbolException [▶ 2401] type exposes the following members.

Events

	Name	Description
	SerializeObjectState	Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)

Reference

SymbolException Class [▶ 2401]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133 TypeAttribute Class

ADS Attribute

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.TypeAttribute
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class TypeAttribute : ITypeAttribute
```

The TypeAttribute type exposes the following members.

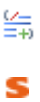
Properties

	Name	Description
	Name [▶ 2414]	Name of the Attribute
	Value [▶ 2414]	Gets the value of the attribute

Methods

	Name	Description
	Equals [▶ 2415]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 2416]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Operators

	Name	Description
	Equality [▶ 2417]	Operator==
	Inequality [▶ 2417]	Implements the != operator.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.1 TypeAttribute Properties

The TypeAttribute [▶ 2413] type exposes the following members.

Properties

	Name	Description
	Name [▶ 2414]	Name of the Attribute
	Value [▶ 2414]	Gets the value of the attribute

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.1.1 TypeAttribute.Name Property

Name of the Attribute

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Name { get; }
```

Property Value

Type: String
The name.

Implements

ITypeAttribute.Name [▶ 2210]

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.1.2 TypeAttribute.Value Property

Gets the value of the attribute

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string Value { get; }
```

Property Value

Type: String
The value.

Implements

ITypeAttribute.Value [▶ 2210]

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.2 TypeAttribute Methods

The TypeAttribute [▶ 2413] type exposes the following members.

Methods

	Name	Description
	Equals [▶ 2415]	Equals (Overrides Object.Equals(Object).)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode [▶ 2416]	Gets the HashCode of the Address (Overrides Object.GetHashCode..)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.2.1 TypeAttribute.Equals Method

Equals

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override bool Equals(
    Object obj
)
```

Parameters

obj	Type: System.Object The object to compare with the current object.

Return Value

Type: Boolean
true if the specified Object is equal to this instance; otherwise, false.

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.2.2 TypeAttribute.GetHashCode Method

Gets the HashCode of the Address

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public override int GetHashCode()
```

Return Value

Type: Int32
A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.3 TypeAttribute Operators

The TypeAttribute [▶ 2413] type exposes the following members.

Operators

	Name	Description
	Equality [▶ 2417]	Operator==
	Inequality [▶ 2417]	Implements the != operator.

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.3.1 TypeAttribute.Equality Operator

Operator==

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator ==(
    TypeAttribute o1,
    TypeAttribute o2
)
```

Parameters

o1	Type: TwinCAT.TypeSystem.TypeAttribute [▶ 2413] The o1.
o2	Type: TwinCAT.TypeSystem.TypeAttribute [▶ 2413] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.133.3.2 TypeAttribute.Inequality Operator

Implements the != operator.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static bool operator !=(
    TypeAttribute o1,
    TypeAttribute o2
)
```

Parameters

o1	Type: TwinCAT.TypeSystem.TypeAttribute [▶ 2413] The o1.
o2	Type: TwinCAT.TypeSystem.TypeAttribute [▶ 2413] The o2.

Return Value

Type: Boolean
The result of the operator.

Reference

TypeAttribute Class [▶ 2413]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134 TypeAttributeCollection Class

Collection of AdsAttributes [▶ 2209]

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.TypeAttributeCollection
Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class TypeAttributeCollection : ITypeAttributeCollection,
    IList<ITypeAttribute>, ICollection<ITypeAttribute>, IEnumerable<ITypeAttribute>,
    IEnumerable
```

The TypeAttributeCollection type exposes the following members.

Constructors

	Name	Description
	TypeAttributeCollection. [▶ 2420]	Initializes a new instance of the TypeAttributeCollection class.

	Name	Description
	TypeAttributeCollection(IEnumerable.ITypeAttribute.) [▶ 2421]	Initializes a new instance of the TypeAttributeCollection class.

Properties

	Name	Description
	Count [▶ 2421]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2422]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2423]	Gets or sets the element at the specified index.
	Item.String. [▶ 2423]	Gets the String with the specified name.

Methods

	Name	Description
	Add [▶ 2425]	Adds an item to the ICollection.T..
	AddRange [▶ 2426]	Adds the range.
	AsReadOnly [▶ 2426]	Gets a read only version of this TypeAttributeCollection
	Clear [▶ 2427]	Removes all items from the ICollection.T..
	Contains(String) [▶ 2428]	Determines whether this TypeAttributeCollection contains the ITypeAttribute [▶ 2209] with the specified name.
	Contains(ITypeAttribute) [▶ 2428]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2429]	Copies to.
S	Empty [▶ 2430]	Returns an Empty TypeAttributeCollection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2430]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2431]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2431]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	Remove(String) [▶ 2432]	Removes the specified ITypeAttribute [▶ 2209] from the TypeAttributeCollection
	Remove(ITypeAttribute) [▶ 2433]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2433]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetAttribute [▶ 2434]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2435]	Tries to get the specified Attribute value.

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.1 TypeAttributeCollection Constructor

Overload List

	Name	Description
	TypeAttributeCollection. [▶ 2420]	Initializes a new instance of the TypeAttributeCollection [▶ 2418] class.
	TypeAttributeCollection(IEnumerable.ITypeAttribute.) [▶ 2421]	Initializes a new instance of the TypeAttributeCollection [▶ 2418] class.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.1.1 TypeAttributeCollection Constructor

Initializes a new instance of the TypeAttributeCollection [▶ 2418] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TypeAttributeCollection()
```

Reference

TypeAttributeCollection Class [▶ 2418]

TypeAttributeCollection Overload [▶ 2420]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.1.2 TypeAttributeCollection Constructor (IEnumerable.ITypeAttribute.)

Initializes a new instance of the TypeAttributeCollection [▶ 2418] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public TypeAttributeCollection(
    IEnumerable<ITypeAttribute> coll
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.ITypeAttribute [▶ 2209]. The coll.

Reference

TypeAttributeCollection Class [▶ 2418]

TypeAttributeCollection Overload [▶ 2420]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.2 TypeAttributeCollection Properties

The TypeAttributeCollection [▶ 2418] type exposes the following members.

Properties

	Name	Description
	Count [▶ 2421]	Gets the number of elements contained in the ICollection.T..
	IsReadOnly [▶ 2422]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2423]	Gets or sets the element at the specified index.
	Item.String. [▶ 2423]	Gets the String with the specified name.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.2.1 TypeAttributeCollection.Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.2.2 TypeAttributeCollection.IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.2.3 TypeAttributeCollection.Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 2423]	Gets or sets the element at the specified index.
	Item.String. [▶ 2423]	Gets the String with the specified name.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ITypeAttribute this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: ITypeAttribute [▶ 2209]
AdsAttribute.

Implements

IList.T..Item.Int32.

Reference

TypeAttributeCollection Class [▶ 2418]

Item Overload [▶ 2423]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Item Property (String)

Gets the String with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public string this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: String
System.String.

Implements

ITypeAttributeCollection.Item.String. [▶ 2213]

Exceptions

Exception	Condition
KeyNotFoundException	

Reference

TypeAttributeCollection Class [▶ 2418]

Item Overload [▶ 2423]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3 TypeAttributeCollection Methods

The TypeAttributeCollection [▶ 2418] type exposes the following members.

Methods

	Name	Description
	Add [▶ 2425]	Adds an item to the ICollection.T..
	AddRange [▶ 2426]	Adds the range.
	AsReadOnly [▶ 2426]	Gets a read only version of this TypeAttributeCollection [▶ 2418]
	Clear [▶ 2427]	Removes all items from the ICollection.T..
	Contains(String) [▶ 2428]	Determines whether this TypeAttributeCollection [▶ 2418] contains the ITypeAttribute [▶ 2209] with the specified name.

	Name	Description
	Contains(ITypeAttribute) [▶ 2428]	Determines whether the ICollection.T. contains a specific value.
	CopyTo [▶ 2429]	Copies to.
S	Empty [▶ 2430]	Returns an Empty TypeAttributeCollection [▶ 2418].
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2430]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2431]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2431]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove(String) [▶ 2432]	Removes the specified ITypeAttribute [▶ 2209] from the TypeAttributeCollection [▶ 2418]
	Remove(ITypeAttribute) [▶ 2433]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2433]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetAttribute [▶ 2434]	Tries to get the specified ITypeAttribute [▶ 2209]
	TryGetValue [▶ 2435]	Tries to get the specified Attribute value.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    ITypeAttribute item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209] The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.2 TypeAttributeCollection.AddRange Method

Adds the range.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void AddRange(
    IEnumerable<ITypeAttribute> items
)
```

Parameters

items	Type: System.Collections.Generic.IEnumerable.ITypeAttribute [▶ 2209]. The items.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.3 TypeAttributeCollection.AsReadOnly Method

Gets a read only version of this TypeAttributeCollection [▶ 2418]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyTypeAttributeCollection AsReadOnly()
```

Field Value

Type: ReadOnlyTypeAttributeCollection [▶ 2340]
As read only.

Return Value

Type: ReadOnlyTypeAttributeCollection [▶ 2340]
ReadOnlyAttributeCollection.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.4 TypeAttributeCollection.Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.5 TypeAttributeCollection.Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2428]	Determines whether this TypeAttributeCollection [▶ 2418] contains the ITypeAttribute [▶ 2209] with the specified name.
	Contains(ITypeAttribute) [▶ 2428]	Determines whether the ICollection.T. contains a specific value.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Contains Method (String)

Determines whether this TypeAttributeCollection [▶ 2418] contains the ITypeAttribute [▶ 2209] with the specified name.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if [contains] [the specified name]; otherwise, false.

Implements

ITypeAttributeCollection.Contains(String) [▶ 2214]

Reference

TypeAttributeCollection Class [▶ 2418]

Contains Overload [▶ 2427]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Contains Method (ITypeAttribute)

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    ITypeAttribute item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209] The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

TypeAttributeCollection Class [▶ 2418]

Contains Overload [▶ 2427]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.6 TypeAttributeCollection.CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    ITypeAttribute[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.ITypeAttribute [▶ 2209]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.7 TypeAttributeCollection.Empty Method

Returns an Empty TypeAttributeCollection [▶ 2418].

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static TypeAttributeCollection Empty()
```

Return Value

Type: TypeAttributeCollection [▶ 2418]
TypeAttributeCollection.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.8 TypeAttributeCollection.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<ITypeAttribute> GetEnumerator()
```

Return Value

Type: IEnumerator.ITypeAttribute [▶ 2209].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.9 TypeAttributeCollection.IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    ITypeAttribute item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209] The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.Insert Method 0

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    ITypeAttribute item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209] The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.Remove Method 1

Overload List

	Name	Description
	Remove(String) [▶ 2432]	Removes the specified ITypeAttribute [▶ 2209] from the TypeAttributeCollection [▶ 2418]
	Remove(ITypeAttribute) [▶ 2433]	Removes the first occurrence of a specific object from the ICollection.T..

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Remove Method (String)

Removes the specified ITypeAttribute [▶ 2209] from the TypeAttributeCollection [▶ 2418]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

TypeAttributeCollection Class [▶ 2418]

Remove Overload [▶ 2432]

TwinCAT.TypeSystem Namespace [▶ 1622]

TypeAttributeCollection.Remove Method (ITypeAttribute)

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    ITypeAttribute item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209] The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

TypeAttributeCollection Class [▶ 2418]

Remove Overload [▶ 2432]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.RemoveAt Method 2

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.TryGetAttribute Method 3

Tries to get the specified ITypeAttribute [▶ 2209]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetAttribute(
    string name,
    out ITypeAttribute att
)
```

Parameters

name	Type: System.String The name of the ITypeAttribute [▶ 2209].
att	Type: TwinCAT.TypeSystem.ITypeAttribute [▶ 2209]. The att.

Return Value

Type: Boolean
true if found, false otherwise.

Implements

ITypeAttributeCollection.TryGetAttribute(String, ITypeAttribute.) [▶ 2215]

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.134.3.1 TypeAttributeCollection.TryGetValue Method 4

Tries to get the specified Attribute value.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetValue(
    string name,
    out string value
)
```

Parameters

name	Type: System.String The name.
value	Type: System.String. The value.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Implements

ITypeAttributeCollection.TryGetValue(String, String.) [▶ 2215]

Reference

TypeAttributeCollection Class [▶ 2418]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135 ValueChangedBaseEventArgs Class

Event args for the RawValueChanged [▶ 2254] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.TypeSystem.ValueChangedBaseEventArgs
      TwinCAT.TypeSystem.RawValueChangedEventArgs [▶ 2289]
      TwinCAT.TypeSystem.ValueChangedEventArgs [▶ 2439]

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ValueChangedBaseEventArgs : EventArgs
```

The ValueChangedBaseEventArgs type exposes the following members.

Constructors

	Name	Description
	ValueChangedBaseEventArgs [▶ 2436]	Initializes a new instance of the RawValueChangedEventArgs [▶ 2289] class.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp
	Symbol [▶ 2438]	Gets the symbol.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135.1 ValueChangedBaseEventArgs Constructor

Initializes a new instance of the RawValueChangedEventArgs [▶ 2289] class.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected ValueChangedBaseEventArgs(
    ISymbol symbol,
    DateTimeOffset timeStamp
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.
timeStamp	Type: System.DateTimeOffset The TwinCAT Real time time stamp (UTC)

Reference

ValueChangedBaseEventArgs Class [▶ 2435]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135.2 ValueChangedBaseEventArgs Properties

The ValueChangedBaseEventArgs [▶ 2435] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp
	Symbol [▶ 2438]	Gets the symbol.

Reference

ValueChangedBaseEventArgs Class [▶ 2435]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135.2.1 ValueChangedBaseEventArgs.DateTime Property

Notification timestamp

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset DateTime { get; }
```

Property Value

Type: DateTimeOffset

Reference

ValueChangedBaseEventArgs Class [▶ 2435]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135.2.2 ValueChangedBaseEventArgs.Symbol Property

Gets the symbol.

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ISymbol Symbol { get; }
```

Property Value

Type: ISymbol [▶ 2176]
The symbol.

Reference

ValueChangedBaseEventArgs Class [▶ 2435]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.135.3 ValueChangedBaseEventArgs Methods

The ValueChangedBaseEventArgs [▶ 2435] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ValueChangedBaseEventArgs Class [▶ 2435]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.136 ValueChangedEventArgs Class

Event args for the ValueChanged [▶ 2269] event.

Inheritance Hierarchy

System.Object
  System.EventArgs
    TwinCAT.TypeSystem.ValueChangedBaseEventArgs [▶ 2435]
      TwinCAT.TypeSystem.ValueChangedEventArgs

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ValueChangedEventArgs : ValueChangedBaseEventArgs
```

The ValueChangedEventArgs type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Symbol [▶ 2438]	Gets the symbol. (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Value [▶ 2440]	The new received Value

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.136.1 ValueChangedEventArgs Properties

The ValueChangedEventArgs [▶ 2439] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2437]	Notification timestamp (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Symbol [▶ 2438]	Gets the symbol. (Inherited from ValueChangedBaseEventArgs [▶ 2435].)
	Value [▶ 2440]	The new received Value

Reference

ValueChangedEventArgs Class [▶ 2439]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.136.1.1 ValueChangedEventArgs.Value Property

The new received Value

Namespace: TwinCAT.TypeSystem [▶ 1622]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object Value { get; }
```

Property Value

Type: Object

Reference

ValueChangedEventArgs Class [▶ 2439]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.11.136.2 ValueChangedEventArgs Methods

The ValueChangedEventArgs [▶ 2439] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ValueChangedEventArgs Class [▶ 2439]

TwinCAT.TypeSystem Namespace [▶ 1622]

6.12 TwinCAT.TypeSystem.Generic Namespace

Namespace for the dynamic part of the common type system.

Classes

	Class	Description
	DataTypeCollection.T. [▶ 2442]	Data type collection
	InstanceCollection.T. [▶ 2460]	Base class for IInstance [▶ 2052] object collections (abstract).
	NamespaceCollection.T. [▶ 2484]	Generic class for Namespace collections
	ReadOnlyDataTypeCollection.T. [▶ 2499]	ReadOnly DataType collection
	ReadOnlyInstanceCollection.T. [▶ 2505]	ReadOnly Instance collection
	ReadOnlyNamespaceCollection.T. [▶ 2515]	Read Only namespace collection
	ReadOnlySymbolCollection.T. [▶ 2522]	Read only symbol collection.
	SymbolCollection.T. [▶ 2526]	Interface represents a collection of ISymbol [▶ 2176] objects.
	SymbolIterator.T. [▶ 2532]	Iterator class for enumerations of Symbols [▶ 2176].

Interfaces

	Interface	Description
	INamespace.TType. [▶ 2458]	Namespace interface
	ISymbolProvider.TNamespace, TDataType, TSymbol. [▶ 2481]	Symbol provider interface

Enumerations

	Enumeration	Description
	SymbolIterationMask [▶ 2532]	Mask Flagset to specify filters for SymbolIterator.T. [▶ 2532].

6.12.1 DataTypeCollection.T. Class

Data type collection

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.DataTypeCollection.T.
    TwinCAT.TypeSystem.DataTypeCollection [▶ 1650]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class DataTypeCollection<T> : IList<T>,
    ICollection<T>, IEnumerable<T>, IEnumerable, IDataTypeCollection<T>
where T : class, IDataType
```

Type Parameters

T

The DataTypeCollection.T. type exposes the following members.

Constructors

	Name	Description
	DataTypeCollection.T.. [▶ 2444]	Initializes a new instance of the DataTypeCollection [▶ 1650] class.
	DataTypeCollection.T.(IEnumerable.T.) [▶ 2444]	Initializes a new instance of the DataTypeCollection.T. class.

Properties

	Name	Description
	Count [▶ 2445]	Gets the count of contained IDataType [▶ 1986]s.
	IsReadOnly [▶ 2446]	Gets a value indicating whether this instance is read only.
	Item.Int32. [▶ 2447]	Gets or sets the IDataType [▶ 1986] at the specified index.
	Item.String. [▶ 2448]	Gets the IDataType [▶ 1986] with the specified name.

Methods

	Name	Description
	Add [▶ 2449]	Adds the specified item to the collection.
	AddRange [▶ 2450]	Adds a range of types
	AsReadOnly [▶ 2450]	Converts the DataTypeCollection.T. into a ReadOnlyCollection.T.
	Clear [▶ 2451]	Clears the collection.
	Clone [▶ 2451]	Clones this instance.
	Contains [▶ 2451]	Determines whether this DataTypeCollection [▶ 1650] contains the specified IDataType [▶ 1986].
	ContainsType [▶ 2452]	Determines whether the container contains the specified IDataType [▶ 1986].
	CopyTo [▶ 2453]	Copies the data types to the specified array, starting at the array index.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2453]	Gets the enumerator.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2454]	Determines the Index of the specified IDataType [▶ 1986].
	Insert [▶ 2454]	Inserts an IDataType [▶ 1986] into the DataTypeCollection [▶ 1650].
	LookupType [▶ 2455]	Determines the specified IDataType [▶ 1986]
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2456]	Removes the specified IDataType [▶ 1986].
	RemoveAt [▶ 2456]	Removes the IDataType [▶ 1986] object at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2457]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995].

Fields

	Name	Description
	list [▶ 2458]	Internal list of data types

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.1 DataTypeCollection.T. Constructor

Overload List

	Name	Description
	DataTypeCollection.T.. [▶ 2444]	Initializes a new instance of the DataTypeCollection [▶ 1650] class.
	DataTypeCollection.T.(IEnumerable.T.) [▶ 2444]	Initializes a new instance of the DataTypeCollection.T. [▶ 2442] class.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.1.1 DataTypeCollection.T. Constructor

Initializes a new instance of the DataTypeCollection [▶ 1650] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection()
```

Reference

DataTypeCollection.T. Class [▶ 2442]

DataTypeCollection.T. Overload [▶ 2444]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.1.2 DataTypeCollection.T. Constructor (IEnumerable.T.)

Initializes a new instance of the DataTypeCollection.T. [▶ 2442] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection(
    IEnumerable<T> types
)
```

Parameters

types	Type: System.Collections.Generic.IEnumerable.T [▶ 2442]. The types.

Reference

DataTypeCollection.T. Class [▶ 2442]

DataTypeCollection.T. Overload [▶ 2444]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.2 DataTypeCollection.T. Properties

The DataTypeCollection.T. [▶ 2442] generic type exposes the following members.

Properties

	Name	Description
	Count [▶ 2445]	Gets the count of contained IDataType [▶ 1986]s.
	IsReadOnly [▶ 2446]	Gets a value indicating whether this instance is read only.
	Item.Int32. [▶ 2447]	Gets or sets the IDataType [▶ 1986] at the specified index.
	Item.String. [▶ 2448]	Gets the IDataType [▶ 1986] with the specified name.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.2.1 DataTypeCollection.T..Count Property

Gets the count of contained IDataType [▶ 1986]s.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.2.2 DataTypeCollection.T..IsReadOnly Property

Gets a value indicating whether this instance is read only.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.2.3 DataTypeCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 2447]	Gets or sets the IDataType [▶ 1986] at the specified index.

	Name	Description
	Item.String. [▶ 2448]	Gets the IDataType [▶ 1986] with the specified name.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

DataTypeCollection.T..Item Property (Int32)

Gets or sets the IDataType [▶ 1986] at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: T [▶ 2442]
T.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

DataTypeCollection.T. Class [▶ 2442]

Item Overload [▶ 2446]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

DataTypeCollection.T..Item Property (String)

Gets the IDataType [▶ 1986] with the specified name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 2442]
T.

Implements

IDataTypeCollection.T..Item.String. [▶ 1996]

Reference

DataTypeCollection.T. Class [▶ 2442]

Item Overload [▶ 2446]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3 DataTypeCollection.T. Methods

The DataTypeCollection.T. [▶ 2442] generic type exposes the following members.

Methods

	Name	Description
	Add [▶ 2449]	Adds the specified item to the collection.
	AddRange [▶ 2450]	Adds a range of types
	AsReadOnly [▶ 2450]	Converts the DataTypeCollection.T. [▶ 2442] into a ReadOnlyCollection.T.
	Clear [▶ 2451]	Clears the collection.
	Clone [▶ 2451]	Clones this instance.
	Contains [▶ 2451]	Determines whether this DataTypeCollection [▶ 1650] contains the specified IDataType [▶ 1986].
	ContainsType [▶ 2452]	Determines whether the container contains the specified IDataType [▶ 1986].

	Name	Description
	CopyTo [▶ 2453]	Copies the data types to the specified array, starting at the array index.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2453]	Gets the enumerator.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2454]	Determines the Index of the specified IDataType [▶ 1986].
	Insert [▶ 2454]	Inserts an IDataType [▶ 1986] into the DataTypeCollection [▶ 1650].
	LookupType [▶ 2455]	Determines the specified IDataType [▶ 1986]
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2456]	Removes the specified IDataType [▶ 1986].
	RemoveAt [▶ 2456]	Removes the IDataType [▶ 1986] object at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2457]	Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995].

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.1 DataTypeCollection.T..Add Method

Adds the specified item to the collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    T item
)
```

Parameters

item	Type: T [▶ 2442] The item.

Implements

ICollection.T..Add(T)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.2 DataTypeCollection.T..AddRange Method

Adds a range of types

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void AddRange(
    IEnumerable<T> types
)
```

Parameters

types	Type: System.Collections.Generic.IEnumerable.T [▶ 2442]. The types.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.3 DataTypeCollection.T..AsReadOnly Method

Converts the DataTypeCollection.T. [▶ 2442] into a ReadOnlyCollection.T.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDataTypeCollection<T> AsReadOnly()
```

Return Value

Type: ReadOnlyDataTypeCollection [▶ 2499].T [▶ 2442].
ReadOnlyDataTypeCollection<T>.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.4 DataTypeCollection.T..Clear Method

Clears the collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.5 DataTypeCollection.T..Clone Method

Clones this instance.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DataTypeCollection<T> Clone()
```

Return Value

Type: DataTypeCollection [▶ 2442].T [▶ 2442].
DataTypeCollection<T>.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.6 DataTypeCollection.T..Contains Method

Determines whether this DataTypeCollection [▶ 1650] contains the specified IDataType [▶ 1986].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    T item
)
```

Parameters

item	Type: T [▶ 2442] The item.

Return Value

Type: Boolean
true if [contains] [the specified item]; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.7 DataTypeCollection.T..ContainsType Method

Determines whether the container contains the specified IDataType [▶ 1986].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsType(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if contained; otherwise, false.

Implements

IDataTypeCollection.T..ContainsType(String) [▶ 1997]

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.8 DataTypeCollection.T..CopyTo Method

Copies the data types to the specified array, starting at the array index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    T[] array,
    int arrayIndex
)
```

Parameters

array	Type: .T [▶ 2442]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.9 DataTypeCollection.T..GetEnumerator Method

Gets the enumerator.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<T> GetEnumerator()
```

Return Value

Type: IEnumerator.T [▶ 2442].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.10 DataTypeCollection.T..IndexOf Method

Determines the Index of the specified IDataType [▶ 1986].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    T item
)
```

Parameters

item	Type: T [▶ 2442] The item.

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.11 DataTypeCollection.T..Insert Method

Inserts an IDataType [▶ 1986] into the DataTypeCollection [▶ 1650].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    T item
)
```

Parameters

index	Type: System.Int32 The index.
item	Type: T [▶ 2442] The item.

Implements

IList.T..Insert(Int32, T)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.12 DataTypeCollection.T..LookupType Method

Determines the specified IDataType [▶ 1986]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T LookupType(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 2442]
The IDataType [▶ 1986] if found, otherwise NULL

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.13 DataTypeCollection.T..Remove Method

Removes the specified IDataType [▶ 1986].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    T item
)
```

Parameters

item	Type: T [▶ 2442] The item.

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.14 DataTypeCollection.T..RemoveAt Method

Removes the IDataType [▶ 1986] object at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The index.

Implements

IList.T..RemoveAt(Int32)

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.3.15 DataTypeCollection.T..TryGetType Method

Tries to get the specified IDataType [▶ 1986] from the IDataTypeCollection.T. [▶ 1995].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetType(
    string name,
    out T type
)
```

Parameters

name	Type: System.String The name.
type	Type: T [▶ 2442]. The type (Out parameter)

Return Value

Type: Boolean
true if found

Implements

IDataTypeCollection.T..TryGetType(String, T.) [▶ 1998]

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.4 DataTypeCollection.T. Fields

The DataTypeCollection.T. [▶ 2442] generic type exposes the following members.

Fields

	Name	Description
	list [▶ 2458]	Internal list of data types

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.1.4.1 DataTypeCollection.T..list Field

Internal list of data types

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected List<T> list
```

Field Value

Type: List.T [▶ 2442].

Reference

DataTypeCollection.T. Class [▶ 2442]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.2 INamespace.TType. Interface

Namespace interface

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface INamespace<TType>
where TType : class, IDataType
```

Type Parameters

TType	DataType class used within this Namespace interface

The INamespace.TType. type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2459]	Data types organized by the INamespace.TType.
	Name [▶ 2460]	Gets the name/ identifier of the Namespace

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.2.1 INamespace.TType. Properties

The INamespace.TType. [▶ 2458] generic type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2459]	Data types organized by the INamespace.TType. [▶ 2458]
	Name [▶ 2460]	Gets the name/ identifier of the Namespace

Reference

INamespace.TType. Interface [▶ 2458]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.2.1.1 INamespace.TType..DataTypes Property

Data types organized by the INamespace.TType. [▶ 2458]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataTypeCollection<TType> DataTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].TType [▶ 2458].
The data types.

Reference

INamespace.TType. Interface [▶ 2458]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.2.1.2 INamespace.TType..Name Property

Gets the name/ identifier of the Namespace

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string Name { get; }
```

Property Value

Type: String
The name.

Reference

INamespace.TType. Interface [▶ 2458]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3 InstanceCollection.T. Class

Base class for IInstance [▶ 2052] object collections (abstract).

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection.T.
    TwinCAT.TypeSystem.FieldCollection [▶ 1945]
    TwinCAT.TypeSystem.Generic.SymbolCollection.T. [▶ 2526]
    TwinCAT.TypeSystem.MemberCollection [▶ 2281]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public abstract class InstanceCollection<T> : IList<T>,
    ICollection<T>, IEnumerable<T>, IEnumerable, IInstanceCollection<T>
where T : class, IInstance
```

Type Parameters

T

The InstanceCollection.T. type exposes the following members.

Constructors

	Name	Description
	InstanceCollection.T .(InstanceCollectionMode) [▶ 2463]	Initializes a new instance of the InstanceCollection.T. class.
	InstanceCollection.T .(IEnumerable.T., InstanceCollectionMode) [▶ 2463]	Initializes a new instance of the InstanceCollection.T. class.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count.
	InnerList [▶ 2465]	Gets the LIst of instances.
	InnerPathDict [▶ 2465]	The Path dictionary
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only.
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index.
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path.
	Mode [▶ 2468]	The mode this InstanceCollection.T. is working in.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item.
	AddRange [▶ 2470]	Adds the specified items to this collection.
	AsReadOnly [▶ 2470]	Converts the InstanceCollection.T. to an ReadOnlyInstanceCollection.T. [▶ 2505]
	Clear [▶ 2471]	Clears this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052]
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name.
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. to the specified array.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetEnumerator [▶ 2474]	Gets the enumerator.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2476]	Gets the name of the instance by.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052].
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item.
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path.
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name.

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.1 InstanceCollection.T. Constructor

Overload List

	Name	Description
	InstanceCollection.T. (InstanceCollectionMode) [▶ 2463]	Initializes a new instance of the InstanceCollection.T. [▶ 2460] class.
	InstanceCollection.T.(IEnumerable.T., InstanceCollectionMode) [▶ 2463]	Initializes a new instance of the InstanceCollection.T. [▶ 2460] class.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.1.1 InstanceCollection.T. Constructor (InstanceCollectionMode)

Initializes a new instance of the InstanceCollection.T. [▶ 2460] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected InstanceCollection(
    InstanceCollectionMode mode
)
```

Parameters

mode	Type: TwinCAT.TypeSystem.InstanceCollectionMode [▶ 2075] The mode.

Reference

InstanceCollection.T. Class [▶ 2460]

InstanceCollection.T. Overload [▶ 2462]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.1.2 InstanceCollection.T. Constructor (IEnumerable.T., InstanceCollectionMode)

Initializes a new instance of the InstanceCollection.T. [▶ 2460] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected InstanceCollection(
    IEnumerable<T> coll,
    InstanceCollectionMode mode
)
```

Parameters

coll	Type: System.Collections.Generic.IEnumerable.T [▶ 2460]. The copy collection
mode	Type: TwinCAT.TypeSystem.InstanceCollectionMode [▶ 2075] The mode.

Reference

InstanceCollection.T. Class [▶ 2460]

InstanceCollection.T. Overload [▶ 2462]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2 InstanceCollection.T. Properties

The InstanceCollection.T. [▶ 2460] generic type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count.
	InnerList [▶ 2465]	Gets the LIst of instances.
	InnerPathDict [▶ 2465]	The Path dictionary
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only.
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index.
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path.
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.1 InstanceCollection.T..Count Property

Gets the collection count.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Property Value

Type: Int32
The count.

Implements

ICollection.T..Count

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.2 InstanceCollection.T..InnerList Property

Gets the LIst of instances.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IList<T> InnerList { get; }
```

Property Value

Type: IList.T [▶ 2460].
The inner list.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.3 InstanceCollection.T..InnerPathDict Property

The Path dictionary

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IDictionary<string, T> InnerPathDict { get; }
```

Property Value

Type: IDictionary.String, T [▶ 2460].

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.4 InstanceCollection.T..IsReadOnly Property

Gets a value indicating whether this instance is read only.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Property Value

Type: Boolean
true if this instance is read only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.5 InstanceCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index.
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

InstanceCollection.T..Item Property (Int32)

Gets or sets the IInstance [▶ 2052] at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: T [▶ 2460]
T.

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

InstanceCollection.T. Class [▶ 2460]

Item Overload [▶ 2466]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

InstanceCollection.T..Item Property (String)

Gets the IInstance [▶ 2052] with the specified instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string instanceSpecifier
] { get; }
```

Parameters

instanceSpecifier	Type: System.String The instance path or Instance Name (dependent of Mode [▶ 2468] setting)

Return Value

Type: T [▶ 2460]
T.

Implements

IInstanceCollection.T..Item.String. [▶ 2059]

Exceptions

Exception	Condition
ArgumentNullException	
ArgumentException	

Remarks

Dependent what this InstanceCollection.T. [▶ 2460] contains configured by the InstanceCollectionMode [▶ 2075] the instance specifier should be the InstanceName [▶ 2054] or the InstancePath [▶ 2055].

Reference

InstanceCollection.T. Class [▶ 2460]

Item Overload [▶ 2466]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.2.6 InstanceCollection.T..Mode Property

The mode this InstanceCollection.T. [▶ 2460] is working in.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public InstanceCollectionMode Mode { get; }
```

Property Value

Type: InstanceCollectionMode [▶ 2075]

Implements

IInstanceCollection.T..Mode [▶ 2060]

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3 InstanceCollection.T. Methods

The InstanceCollection.T. [▶ 2460] generic type exposes the following members.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item.

	Name	Description
	AddRange [▶ 2470]	Adds the specified items to this collection.
	AsReadOnly [▶ 2470]	Converts the InstanceCollection.T. [▶ 2460] to an ReadOnlyInstanceCollection.T. [▶ 2505]
	Clear [▶ 2471]	Clears this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052]
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name.
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2476]	Gets the name of the instance by.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052].
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item.
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path.
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.1 InstanceCollection.T..Add Method

Adds the specified item.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    T item
)
```

Parameters

item	Type: T [▶ 2460] The item.

Implements

ICollection.T..Add(T)

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.2 InstanceCollection.T..AddRange Method

Adds the specified items to this collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void AddRange(
    IEnumerable<T> items
)
```

Parameters

items	Type: System.Collections.Generic.IEnumerable.T [▶ 2460]. The items.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.3 InstanceCollection.T..AsReadOnly Method

Converts the InstanceCollection.T. [▶ 2460] to an ReadOnlyInstanceCollection.T. [▶ 2505]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyInstanceCollection<T> AsReadOnly()
```

Return Value

Type: ReadOnlyInstanceCollection [▶ 2505].T [▶ 2460].
ReadOnlyInstanceCollection<T>.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.4 InstanceCollection.T..Clear Method

Clears this instance.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.5 InstanceCollection.T..Contains Method

Overload List

	Name	Description
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052]

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

InstanceCollection.T..Contains Method (String)

Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string instanceSpecifier
)
```

Parameters

instanceSpecifier	Type: System.String The instance path or Instance Name (dependent of Mode [▶ 2468] setting)

Return Value

Type: Boolean
true if [contains] [the specified instance path]; otherwise, false.

Implements

IInstanceCollection.T..Contains(String) [▶ 2061]

Exceptions

Exception	Condition
ArgumentNullException	instancePath
ArgumentException	

Reference

InstanceCollection.T. Class [▶ 2460]

Contains Overload [▶ 2471]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

InstanceCollection.T..Contains Method (T)

Determines whether this collection contains the specified IInstance [▶ 2052]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    T item
)
```

Parameters

item	Type: T [▶ 2460] The item.

Return Value

Type: Boolean
true if [contains] [the specified item]; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

InstanceCollection.T. Class [▶ 2460]

Contains Overload [▶ 2471]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.6 InstanceCollection.T..ContainsName Method

Determines whether the specified instance name contains name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: Boolean
true if the specified instance name contains name; otherwise, false.

Implements

IInstanceCollection.T..ContainsName(String) [▶ 2062]

Exceptions

Exception	Condition
NotImplementedException	

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.7 InstanceCollection.T..CopyTo Method

Copies this InstanceCollection.T. [▶ 2460] to the specified array.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    T[] array,
    int arrayIndex
)
```

Parameters

array	Type: .T [▶ 2460]. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.8 InstanceCollection.T..GetEnumerator Method

Gets the enumerator.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<T> GetEnumerator()
```

Return Value

Type: IEnumerator.T [▶ 2460].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.9 InstanceCollection.T..GetInstance Method

Gets the IInstance [▶ 2052]by instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T GetInstance(
    string instanceSpecifier
)
```

Parameters

instanceSpecifier	Type: System.String The instance path or Instance Name (dependent of Mode [▶ 2468] setting)

Return Value

Type: T [▶ 2460]
T.

Implements

IInstanceCollection.T..GetInstance(String) [▶ 2062]

Exceptions

Exception	Condition
ArgumentException	Path not found!;instancePath

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.10 InstanceCollection.T..GetInstanceByName Method

Gets the name of the instance by.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IList<T> GetInstanceByName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: IList.T [▶ 2460].
IList<T>.

Implements

IInstanceCollection.T..GetInstanceByName(String) [▶ 2063]

Exceptions

Exception	Condition
ArgumentException	Name not found!;instanceName

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.11 InstanceCollection.T..IndexOf Method

Determines the index of the specified IInstance [▶ 2052].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    T item
)
```

Parameters

item	Type: T [▶ 2460] The item.

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.12 InstanceCollection.T..Insert Method

Inserts the specified IInstance [▶ 2052] at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    T instance
)
```

Parameters

index	Type: System.Int32 The instance.
instance	Type: T [▶ 2460] The item.

Implements

IList.T..Insert(Int32, T)

Exceptions

Exception	Condition
ArgumentOutOfRangeException	index or index
ArgumentNullException	

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.13 InstanceCollection.T..Remove Method

Removes the specified item.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    T item
)
```

Parameters

item	Type: T [▶ 2460] The item.

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.14 InstanceCollection.T..RemoveAt Method

Removes the IInstance [▶ 2052] at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The index.

Implements

IList.T..RemoveAt(Int32)

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.15 InstanceCollection.T..TryGetInstance Method

Tries to get the IInstance [▶ 2052]. of the specified path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInstance(
    string instanceSpecifier,
    out T symbol
)
```

Parameters

instanceSpecifier	Type: System.String The instance path or Instance Name (dependent of Mode [▶ 2468] setting)
symbol	Type: T [▶ 2460]. The symbol.

Return Value

Type: Boolean
true if the IInstance [▶ 2052] is found; otherwise, false

Implements

IInstanceCollection.T..TryGetInstance(String, T.) [▶ 2063]

Exceptions

Exception	Condition
ArgumentNullException	instancePath
ArgumentException	

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.3.3.16 InstanceCollection.T..TryGetInstanceByName Method

Tries to get Instances by name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public virtual bool TryGetInstanceByName(
    string instanceName,
    out IList<T> instances
)
```

Parameters

instanceName	Type: System.String Name of the instance.
instances	Type: System.Collections.Generic.IList.T [▶ 2460].. The instances found.

Return Value

Type: Boolean
true if the IInstance [▶ 2052] is found; otherwise, false

Implements

IInstanceCollection.T..TryGetInstanceByName(String, IList.T..) [▶ 2064]

Reference

InstanceCollection.T. Class [▶ 2460]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4 ISymbolProvider.TNamespace, TDataType, TSymbol. Interface

Symbol provider interface

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface ISymbolProvider<TNamespace, TDataType, TSymbol>
where TDataType : class, IDataType
where TSymbol : class, ISymbol
```

Type Parameters

TNamespace	Namespace type
TDataType	DataType type
TSymbol	Symbol type

The ISymbolProvider.TNamespace, TDataType, TSymbol. type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2482]	Gets all data types from all Namespaces
	Namespaces [▶ 2482]	Get the Namespaces of DataTypes for this Symbol provider
	RootNamespace [▶ 2483]	Gets the root (main) namespace of the Symbol provider.
	RootNamespaceName [▶ 2483]	Gets the name of the root namespace
	Symbols [▶ 2484]	Gets the (root) symbols of the Symbol provider.

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1 ISymbolProvider.TNamespace, TDataType, TSymbol. Properties

The ISymbolProvider.TNamespace, TDataType, TSymbol. [▶ 2481] generic type exposes the following members.

Properties

	Name	Description
	DataTypes [▶ 2482]	Gets all data types from all Namespaces
	Namespaces [▶ 2482]	Get the Namespaces of DataTypes for this Symbol provider

	Name	Description
	RootNamespace [▶ 2483]	Gets the root (main) namespace of the Symbol provider.
	RootNamespaceName [▶ 2483]	Gets the name of the root namespace
	Symbols [▶ 2484]	Gets the (root) symbols of the Symbol provider.

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1.1 ISymbolProvider.TNamespace, TDataType, TSymbol..DataTypes Property

Gets all data types from all Namespaces

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IDataTypeCollection<TDataType> DataTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].TDataType [▶ 2481].
The data types.

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1.2 ISymbolProvider.TNamespace, TDataType, TSymbol..Namespaces Property

Get the Namespaces of DataTypes for this Symbol provider

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
INamespaceCollection<TDataType> Namespaces { get; }
```

Property Value

Type: INamespaceCollection [▶ 2073].TDataType [▶ 2481].
ReadOnly collection of the namespaces.

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1.3 ISymbolProvider.TNamespace, TDataType, TSymbol..RootNamespace Property

Gets the root (main) namespace of the Symbol provider.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
INamespace<TDataType> RootNamespace { get; }
```

Property Value

Type: INamespace [▶ 2458].TDataType [▶ 2481].
The root namespace.

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1.4 ISymbolProvider.TNamespace, TDataType, TSymbol..RootNamespaceName Property

Gets the name of the root namespace

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
string RootNamespaceName { get; }
```

Property Value

Type: String
The namespace.

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.4.1.5 ISymbolProvider.TNamespace, TDataType, TSymbol..Symbols Property

Gets the (root) symbols of the Symbol provider.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
ISymbolCollection<TSymbol> Symbols { get; }
```

Property Value

Type: ISymbolCollection [▶ 2185].TSymbol [▶ 2481].
Read only collection of the Symbols

Reference

ISymbolProvider.TNamespace, TDataType, TSymbol. Interface [▶ 2481]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5 NamespaceCollection.T. Class

Generic class for Namespace collections

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.NamespaceCollection.T.
Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class NamespaceCollection<T> : IList<INamespace<T>>,
    ICollection<INamespace<T>>, IEnumerable<INamespace<T>>, IEnumerable
where T : class, IDataType
```

Type Parameters

T

The NamespaceCollection.T. type exposes the following members.

Constructors

	Name	Description
	NamespaceCollection.T. [▶ 2486]	Initializes a new instance of the NamespaceCollection.T. class.

Properties

	Name	Description
	AllTypes [▶ 2487]	Gets all types included in all namespaces.
	Count [▶ 2487]	Gets the number of elements contained in the ICollection.T..
	InnerAllTypes [▶ 2488]	Dictionary FullPath -> IDataType
	IsReadOnly [▶ 2488]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2489]	Gets or sets the element at the specified index.
	Item.String. [▶ 2490]	Gets or sets the element at the specified index.

Methods

	Name	Description
	Add [▶ 2491]	Adds an item to the ICollection.T..
	Clear [▶ 2492]	Removes all items from the ICollection.T..
	Contains [▶ 2492]	Determines whether the ICollection.T. contains a specific value.
	ContainsNamespace [▶ 2493]	Determines whether the specified name contains namespace.
	CopyTo [▶ 2493]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2494]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2494]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2495]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2496]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2496]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

	Name	Description
	TryGetNamespace [▶ 2497]	Tries to get the namespace object
	TryGetType [▶ 2497]	Tries to get the specified type.
	TryGetTypeByFullName [▶ 2498]	Tries to get the data type by full name.

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.1 NamespaceCollection.T. Constructor

Initializes a new instance of the NamespaceCollection.T. [▶ 2484] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public NamespaceCollection()
```

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2 NamespaceCollection.T. Properties

The NamespaceCollection.T. [▶ 2484] generic type exposes the following members.

Properties

	Name	Description
	AllTypes [▶ 2487]	Gets all types included in all namespaces.
	Count [▶ 2487]	Gets the number of elements contained in the ICollection.T..
	InnerAllTypes [▶ 2488]	Dictionary FullPath -> IDataType
	IsReadOnly [▶ 2488]	Gets a value indicating whether the ICollection.T. is read-only.
	Item.Int32. [▶ 2489]	Gets or sets the element at the specified index.
	Item.String. [▶ 2490]	Gets or sets the element at the specified index.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2.1 NamespaceCollection.T..AllTypes Property

Gets all types included in all namespaces.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataTypeCollection<T> AllTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].T [▶ 2484].
All types.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2.2 NamespaceCollection.T..Count Property

Gets the number of elements contained in the ICollection.T..

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int Count { get; }
```

Return Value

Type: Int32
The number of elements contained in the ICollection.T..

Implements

ICollection.T..Count

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2.3 NamespaceCollection.T..InnerAllTypes Property

Dictionary FullPath -> IDataType

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
protected IDictionary<string, T> InnerAllTypes { get; }
```

Property Value

Type: IDictionary.String, T [▶ 2484].

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2.4 NamespaceCollection.T..IsReadOnly Property

Gets a value indicating whether the ICollection.T. is read-only.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool IsReadOnly { get; }
```

Return Value

Type: Boolean
true if the ICollection.T. is read-only; otherwise, false.

Implements

ICollection.T..IsReadOnly

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.2.5 NamespaceCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32. [▶ 2489]	Gets or sets the element at the specified index.
	Item.String. [▶ 2490]	Gets or sets the element at the specified index.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

NamespaceCollection.T..Item Property (Int32)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public INamespace<T> this[
    int index
] { get; set; }
```

Parameters

index	Type: System.Int32 The index.

Return Value

Type: INamespace [▶ 2458].T [▶ 2484].

Implements

IList.T..Item.Int32.

Exceptions

Exception	Condition
NotImplementedException	

Reference

NamespaceCollection.T. Class [▶ 2484]

Item Overload [▶ 2489]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

NamespaceCollection.T..Item Property (String)

Gets or sets the element at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public INamespace<T> this[
    string str
] { get; }
```

Parameters

str	Type: System.String The STR.

Return Value

Type: INamespace [▶ 2458].T [▶ 2484].

Reference

NamespaceCollection.T. Class [▶ 2484]

Item Overload [▶ 2489]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3 NamespaceCollection.T. Methods

The NamespaceCollection.T. [▶ 2484] generic type exposes the following members.

Methods

	Name	Description
	Add [▶ 2491]	Adds an item to the ICollection.T..
	Clear [▶ 2492]	Removes all items from the ICollection.T..
	Contains [▶ 2492]	Determines whether the ICollection.T. contains a specific value.
	ContainsNamespace [▶ 2493]	Determines whether the specified name contains namespace.
	CopyTo [▶ 2493]	Copies to.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)

	Name	Description
	GetEnumerator [▶ 2494]	Returns an enumerator that iterates through the collection.
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2494]	Determines the index of a specific item in the IList.T..
	Insert [▶ 2495]	Inserts an item to the IList.T. at the specified index.
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2496]	Removes the first occurrence of a specific object from the ICollection.T..
	RemoveAt [▶ 2496]	Removes the IList.T. item at the specified index.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetNamespace [▶ 2497]	Tries to get the namespace object
	TryGetType [▶ 2497]	Tries to get the specified type.
	TryGetTypeByFullName [▶ 2498]	Tries to get the data type by full name.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.1 NamespaceCollection.T..Add Method

Adds an item to the ICollection.T..

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Add(
    INamespace<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484]. The object to add to the ICollection.T..

Implements

ICollection.T..Add(T)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.2 NamespaceCollection.T..Clear Method

Removes all items from the ICollection.T..

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Clear()
```

Implements

ICollection.T..Clear.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.3 NamespaceCollection.T..Contains Method

Determines whether the ICollection.T. contains a specific value.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    INamespace<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484]. The object to locate in the ICollection.T..

Return Value

Type: Boolean
true if item is found in the ICollection.T.; otherwise, false.

Implements

ICollection.T..Contains(T)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.4 NamespaceCollection.T..ContainsNamespace Method

Determines whether the specified name contains namespace.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsNamespace(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if the specified name contains namespace; otherwise, false.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.5 NamespaceCollection.T..CopyTo Method

Copies to.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void CopyTo(
    INamespace<T>[] array,
    int arrayIndex
)
```

Parameters

array	Type: .TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484].. The array.
arrayIndex	Type: System.Int32 Index of the array.

Implements

ICollection.T..CopyTo(.T., Int32)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.6 NamespaceCollection.T..GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<INamespace<T>> GetEnumerator()
```

Return Value

Type: IEnumerator.INamespace [▶ 2458].T [▶ 2484]..
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.7 NamespaceCollection.T..IndexOf Method

Determines the index of a specific item in the IList.T..

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int IndexOf(
    INamespace<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484]. The object to locate in the IList.T..

Return Value

Type: Int32
The index of item if found in the list; otherwise, -1.

Implements

IList.T..IndexOf(T)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.8 NamespaceCollection.T..Insert Method

Inserts an item to the IList.T. at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void Insert(
    int index,
    INamespace<T> item
)
```

Parameters

index	Type: System.Int32 The zero-based index at which item should be inserted.
item	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484]. The object to insert into the IList.T..

Implements

IList.T..Insert(Int32, T)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.9 NamespaceCollection.T..Remove Method

Removes the first occurrence of a specific object from the ICollection.T..

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Remove(
    INamespace<T> item
)
```

Parameters

item	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484]. The object to remove from the ICollection.T..

Return Value

Type: Boolean
true if item was successfully removed from the ICollection.T.; otherwise, false. This method also returns false if item is not found in the original ICollection.T..

Implements

ICollection.T..Remove(T)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.10 NamespaceCollection.T..RemoveAt Method

Removes the IList.T. item at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void RemoveAt(
    int index
)
```

Parameters

index	Type: System.Int32 The zero-based index of the item to remove.

Implements

IList.T..RemoveAt(Int32)

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.11 NamespaceCollection.T..TryGetNamespace Method

Tries to get the namespace object

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetNamespace(
    string name,
    out INamespace<T> nspace
)
```

Parameters

name	Type: System.String The name.
nspace	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2484].. The namespace object (out-parameter)

Return Value

Type: Boolean
true if found, false if not contained.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.12 NamespaceCollection.T..TryGetType Method

Tries to get the specified type.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetType(
    string typeName,
    out T dataType
)
```

Parameters

typeName	Type: System.String Data type name
dataType	Type: T [▶ 2484]. The found data type (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Exceptions

Exception	Condition
ArgumentNullException	typeName
ArgumentException	

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.5.3.13 NamespaceCollection.T..TryGetTypeByFullName Method

Tries to get the data type by full name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetTypeByFullName(
    string fullname,
    out T dataType
)
```

Parameters

fullname	Type: System.String DataTypes full name.
dataType	Type: T [▶ 2484]. Found data type (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Reference

NamespaceCollection.T. Class [▶ 2484]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6 ReadOnlyDataTypeCollection.T. Class

ReadOnly DataType collection

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.T.
    TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection.T.
      TwinCAT.TypeSystem.ReadOnlyDataTypeCollection [▶ 2291]
Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyDataTypeCollection<T> : ReadOnlyCollection<T>,
    IDataTypeCollection<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : class, IDataType
```

Type Parameters

T

The ReadOnlyDataTypeCollection.T. type exposes the following members.

Constructors

	Name	Description
	ReadOnlyDataTypeCollection.T. (DataTypeCollection.T.) [▶ 2501]	Initializes a new instance of the ReadOnlyDataTypeCollection.T. class.
	ReadOnlyDataTypeCollection.T. (ReadOnlyDataTypeCollection.T.) [▶ 2501]	Initializes a new instance of the ReadOnlyDataTypeCollection.T. class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T..)

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T..)
	Item.String. [▶ 2502]	Gets the element with the specified type name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T..)

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	ContainsType [▶ 2504]	Determines whether the specified name contains type.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2505]	Tries to get the Type with the specified name out of the collection.

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.1 ReadOnlyDataTypeCollection.T. Constructor

Overload List

	Name	Description
	ReadOnlyDataTypeCollection.T. (DataTypeCollection.T.) [▶ 2501]	Initializes a new instance of the ReadOnlyDataTypeCollection.T. [▶ 2499] class.

	Name	Description
	ReadOnlyDataTypeCollection.T. (ReadOnlyDataTypeCollection.T.) [▶ 2501]	Initializes a new instance of the ReadOnlyDataTypeCollection.T. [▶ 2499] class.

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.1.1 ReadOnlyDataTypeCollection.T. Constructor (DataTypeCollection.T.)

Initializes a new instance of the ReadOnlyDataTypeCollection.T. [▶ 2499] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDataTypeCollection(
    DataTypeCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.Generic.DataTypeCollection [▶ 2442].T [▶ 2499]. The collection.

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

ReadOnlyDataTypeCollection.T. Overload [▶ 2500]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.1.2 ReadOnlyDataTypeCollection.T. Constructor (ReadOnlyDataTypeCollection.T.)

Initializes a new instance of the ReadOnlyDataTypeCollection.T. [▶ 2499] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyDataTypeCollection(
    ReadOnlyDataTypeCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.Generic.ReadOnlyDataTypeCollection [▶ 2499].T [▶ 2499]. The coll.

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

ReadOnlyDataTypeCollection.T. Overload [▶ 2500]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.2 ReadOnlyDataTypeCollection.T. Properties

The ReadOnlyDataTypeCollection.T. [▶ 2499] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	Item.String. [▶ 2502]	Gets the element with the specified type name.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T [▶ 2499]..)

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.2.1 ReadOnlyDataTypeCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	Item.String. [▶ 2502]	Gets the element with the specified type name.

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

ReadOnlyDataTypeCollection.T..Item Property (String)

Gets the element with the specified type name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: T [▶ 2499]
T.

Implements

IDataTypeCollection.T..Item.String. [▶ 1996]

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

Item Overload [▶ 2502]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.3 ReadOnlyDataTypeCollection.T. Methods

The ReadOnlyDataTypeCollection.T. [▶ 2499] generic type exposes the following members.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	ContainsType [▶ 2504]	Determines whether the specified name contains type.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2499]..)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2499]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetType [▶ 2505]	Tries to get the Type with the specified name out of the collection.

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.3.1 ReadOnlyDataTypeCollection.T..ContainsType Method

Determines whether the specified name contains type.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsType(
    string name
)
```

Parameters

name	Type: System.String The name.

Return Value

Type: Boolean
true if the specified name contains type; otherwise, false.

Implements

IDataTypeCollection.T..ContainsType(String) [▶ 1997]

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.6.3.2 ReadOnlyDataTypeCollection.T..TryGetType Method

Tries to get the Type with the specified name out of the collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetType(
    string name,
    out T type
)
```

Parameters

name	Type: System.String The name.
type	Type: T [▶ 2499]. The type.

Return Value

Type: Boolean
true if found

Implements

IDataTypeCollection.T..TryGetType(String, T.) [▶ 1998]

Reference

ReadOnlyDataTypeCollection.T. Class [▶ 2499]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7 ReadOnlyInstanceCollection.T. Class

ReadOnly Instance collection

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.T.
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection.T.
      TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection.T. [▶ 2522]
      TwinCAT.TypeSystem.ReadOnlyFieldCollection [▶ 2317]
      TwinCAT.TypeSystem.ReadOnlyMemberCollection [▶ 2322]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyInstanceCollection<T> : ReadOnlyCollection<T>,
    IInstanceCollection<T>, IList<T>, ICollection<T>, IEnumerable<T>,
    IEnumerable
where T : class, IInstance
```

Type Parameters

T

The ReadOnlyInstanceCollection.T. type exposes the following members.

Constructors

	Name	Description
	ReadOnlyInstanceCollection.T. [▶ 2507]	Initializes a new instance of the ReadOnlyInstanceCollection.T. class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075].

Methods

	Name	Description
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. contains an instance with the specified instance path.
	ContainsName [▶ 2511]	Determines whether the specified instance is contained.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)

	Name	Description
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path.
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name.

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.1 ReadOnlyInstanceCollection.T. Constructor

Initializes a new instance of the ReadOnlyInstanceCollection.T. [▶ 2505] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyInstanceCollection(
    IInstanceCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].T [▶ 2505]. The coll.

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.2 ReadOnlyInstanceCollection.T. Properties

The ReadOnlyInstanceCollection.T. [▶ 2505] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075].

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.2.1 ReadOnlyInstanceCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path.

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

ReadOnlyInstanceCollection.T..Item Property (String)

Gets the element with the specified instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T this[
    string instancePath
] { get; }
```

Parameters

instancePath	Type: System.String The instance path.

Return Value

Type: T [▶ 2505]
The instance if contained.

Implements

IInstanceCollection.T..Item.String. [▶ 2059]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

Item Overload [▶ 2508]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.2.2 ReadOnlyInstanceCollection.T..Mode Property

Gets the InstanceCollectionMode [▶ 2075].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public InstanceCollectionMode Mode { get; }
```

Property Value

Type: InstanceCollectionMode [▶ 2075]
The mode.

Implements

IInstanceCollection.T..Mode [▶ 2060]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3 ReadOnlyInstanceCollection.T. Methods

The ReadOnlyInstanceCollection.T. [▶ 2505] generic type exposes the following members.

Methods

	Name	Description
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2505]..)

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path.
	ContainsName [▶ 2511]	Determines whether the specified instance is contained.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path.
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name.
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path.
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name.

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.1 ReadOnlyInstanceCollection.T..Contains Method

Overload List

	Name	Description
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2505]..)
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path.

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

ReadOnlyInstanceCollection.T..Contains Method (String)

Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Contains(
    string instancePath
)
```

Parameters

instancePath	Type: System.String The instance path.

Return Value

Type: Boolean
true if contains the specified instance path; otherwise, false.

Implements

IInstanceCollection.T..Contains(String) [▶ 2061]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

Contains Overload [▶ 2510]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.2 ReadOnlyInstanceCollection.T..ContainsName Method

Determines whether the specified instance is contained.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: Boolean
true, if instance name is found.

Implements

IInstanceCollection.T..ContainsName(String) [▶ 2062]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.3 ReadOnlyInstanceCollection.T..GetInstance Method

Gets the IInstance [▶ 2052]by instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T GetInstance(
    string instancePath
)
```

Parameters

instancePath	Type: System.String The instance path.

Return Value

Type: T [▶ 2505]
T.

Implements

IInstanceCollection.T..GetInstance(String) [▶ 2062]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.4 ReadOnlyInstanceCollection.T..GetInstanceByName Method

Gets the IInstance [▶ 2052] by instance name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IList<T> GetInstanceByName(
    string instanceName
)
```

Parameters

instanceName	Type: System.String Name of the instance.

Return Value

Type: IList.T [▶ 2505].
IList<T>.

Implements

IInstanceCollection.T..GetInstanceByName(String) [▶ 2063]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.5 ReadOnlyInstanceCollection.T..TryGetInstance Method

Tries to get the instance with the specified instance path.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInstance(
    string instancePath,
    out T instance
)
```

Parameters

instancePath	Type: System.String The instance path.
instance	Type: T [▶ 2505]. The instance.

Return Value

Type: Boolean
true, if found, false if not contained.

Implements

IInstanceCollection.T..TryGetInstance(String, T.) [▶ 2063]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.7.3.6 ReadOnlyInstanceCollection.T..TryGetInstanceByName Method

Tries to get the instance by name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInstanceByName(
    string instanceName,
    out IList<T> symbols
)
```

Parameters

instanceName	Type: System.String Name of the instance.
symbols	Type: System.Collections.Generic.IList.T [▶ 2505].. The found symbols (out-parameter)

Return Value

Type: Boolean
true, if found; false if not contained.

Implements

IInstanceCollection.T..TryGetInstanceByName(String, IList.T..) [▶ 2064]

Reference

ReadOnlyInstanceCollection.T. Class [▶ 2505]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8 ReadOnlyNamespaceCollection.T. Class

Read Only namespace collection

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.INamespace [▶ 2458].T..
    TwinCAT.TypeSystem.Generic.ReadOnlyNamespaceCollection.T.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlyNamespaceCollection<T> : ReadOnlyCollection<INamespace<T>>,
    INamespaceCollection<T>, ICollection<INamespace<T>>, IEnumerable<INamespace<T>>,
    IEnumerable
where T : class, IDataType
```

Type Parameters

T

The ReadOnlyNamespaceCollection.T. type exposes the following members.

Constructors

	Name	Description
	ReadOnlyNamespaceCollection.T. [▶ 2516]	Initializes a new instance of the ReadOnlyNamespaceCollection.T. class.

Properties

	Name	Description
	AllTypes [▶ 2517]	Gets all types included in all namespaces.
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	Item.String. [▶ 2518]	Gets the element at the specified index.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	ContainsNamespace [▶ 2519]	Determines whether this collection contains a namespace with the specified name.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T...)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetNamespace [▶ 2520]	Tries to get the namespace with the specified name.
	TryGetType [▶ 2521]	Tries to get the specified data type.
	TryGetTypeByFullName [▶ 2521]	Tries to get the specified type (by fullName)

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.1 ReadOnlyNamespaceCollection.T. Constructor

Initializes a new instance of the ReadOnlyNamespaceCollection.T. [▶ 2515] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlyNamespaceCollection(
    NamespaceCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.Generic.NamespaceCollection [▶ 2484].T [▶ 2515]. The coll.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.2 ReadOnlyNamespaceCollection.T. Properties

The ReadOnlyNamespaceCollection.T. [▶ 2515] generic type exposes the following members.

Properties

	Name	Description
	AllTypes [▶ 2517]	Gets all types included in all namespaces.
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	Item.String. [▶ 2518]	Gets the element at the specified index.
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.2.1 ReadOnlyNamespaceCollection.T..AllTypes Property

Gets all types included in all namespaces.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IDataTypeCollection<T> AllTypes { get; }
```

Property Value

Type: IDataTypeCollection [▶ 1995].T [▶ 2515].
All types.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.2.2 ReadOnlyNamespaceCollection.T..Item Property

Overload List

	Name	Description
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	Item.String. [▶ 2518]	Gets the element at the specified index.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

ReadOnlyNamespaceCollection.T..Item Property (String)

Gets the element at the specified index.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public INamespace<T> this[
    string name
] { get; }
```

Parameters

name	Type: System.String The name.

Return Value

Type: INamespace [▶ 2458].T [▶ 2515].

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

Item Overload [▶ 2518]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.3 ReadOnlyNamespaceCollection.T. Methods

The ReadOnlyNamespaceCollection.T. [▶ 2515] generic type exposes the following members.

Methods

	Name	Description
	Contains	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	ContainsNamespace [▶ 2519]	Determines whether this collection contains a namespace with the specified name.
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.INamespace [▶ 2458].T [▶ 2515]...)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetNamespace [▶ 2520]	Tries to get the namespace with the specified name.
	TryGetType [▶ 2521]	Tries to get the specified data type.
	TryGetTypeByFullName [▶ 2521]	Tries to get the specified type (by fullName)

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.3.1 ReadOnlyNamespaceCollection.T..ContainsNamespace Method

Determines whether this collection contains a namespace with the specified name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool ContainsNamespace(
    string name
)
```

Parameters

name	Type: System.String The name of the namespace

Return Value

Type: Boolean
true if the namespace is contained; otherwise, false.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.3.2 ReadOnlyNamespaceCollection.T..TryGetNamespace Method

Tries to get the namespace with the specified name.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetNamespace(
    string name,
    out INamespace<T> nspace
)
```

Parameters

name	Type: System.String Namespace name.
nspace	Type: TwinCAT.TypeSystem.Generic.INamespace [▶ 2458].T [▶ 2515].. The found namespace (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.3.3 ReadOnlyNamespaceCollection.T..TryGetType Method

Tries to get the specified data type.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetType(
    string typeName,
    out T dataType
)
```

Parameters

typeName	Type: System.String Name of the type.
dataType	Type: T [▶ 2515]. Data Type (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.8.3.4 ReadOnlyNamespaceCollection.T..TryGetTypeByFullName Method

Tries to get the specified type (by fullName)

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetTypeByFullName(
    string fullname,
    out T dataType
)
```

Parameters

fullname	Type: System.String FullName of the data type.
dataType	Type: T [▶ 2515]. Found Data type (out-parameter).

Return Value

Type: Boolean
true if found, false if not contained.

Reference

ReadOnlyNamespaceCollection.T. Class [▶ 2515]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.9 ReadOnlySymbolCollection.T. Class

Read only symbol collection.

Inheritance Hierarchy

System.Object
  System.Collections.ObjectModel.ReadOnlyCollection.T.
    TwinCAT.TypeSystem.Generic.ReadOnlyInstanceCollection [▶ 2505].T.
      TwinCAT.TypeSystem.Generic.ReadOnlySymbolCollection.T.
        TwinCAT.TypeSystem.ReadOnlySymbolCollection [▶ 2336]
Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ReadOnlySymbolCollection<T> : ReadOnlyInstanceCollection<T>
where T : class, ISymbol
```

Type Parameters

T

The ReadOnlySymbolCollection.T. type exposes the following members.

Constructors

	Name	Description
	ReadOnlySymbolCollection.T. [▶ 2524]	Initializes a new instance of the ReadOnlySymbolCollection.T. class.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T..)

	Name	Description
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T..)
	Empty [▶ 2525]	Returns an empty collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.9.1 ReadOnlySymbolCollection.T. Constructor

Initializes a new instance of the ReadOnlySymbolCollection.T. [▶ 2522] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlySymbolCollection(
    IInstanceCollection<T> coll
)
```

Parameters

coll	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].T [▶ 2522]. The coll.

Reference

ReadOnlySymbolCollection.T. Class [▶ 2522]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.9.2 ReadOnlySymbolCollection.T. Properties

The ReadOnlySymbolCollection.T. [▶ 2522] generic type exposes the following members.

Properties

	Name	Description
	Count	Gets the number of elements contained in the ReadOnlyCollection.T. instance. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	Item.String. [▶ 2508]	Gets the element with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Item.Int32.	Gets the element at the specified index. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	Items	Returns the IList.T. that the ReadOnlyCollection.T. wraps. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	Mode [▶ 2509]	Gets the InstanceCollectionMode [▶ 2075]. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlySymbolCollection.T. Class [▶ 2522]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.9.3 ReadOnlySymbolCollection.T. Methods

The ReadOnlySymbolCollection.T. [▶ 2522] generic type exposes the following members.

Methods

	Name	Description
	Contains(String) [▶ 2511]	Determines whether the ReadOnlyInstanceCollection.T. [▶ 2505] contains an instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	Contains(T)	Determines whether an element is in the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	ContainsName [▶ 2511]	Determines whether the specified instance is contained. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	CopyTo	Copies the entire ReadOnlyCollection.T. to a compatible one-dimensional Array, starting at the specified index of the target array. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
S	Empty [▶ 2525]	Returns an empty collection.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator	Returns an enumerator that iterates through the ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2512]	Gets the IInstance [▶ 2052]by instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetInstanceByName [▶ 2513]	Gets the IInstance [▶ 2052] by instance name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf	Searches for the specified object and returns the zero-based index of the first occurrence within the entire ReadOnlyCollection.T.. (Inherited from ReadOnlyCollection.T [▶ 2522]..)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2513]	Tries to get the instance with the specified instance path. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)
	TryGetInstanceByName [▶ 2514]	Tries to get the instance by name. (Inherited from ReadOnlyInstanceCollection.T. [▶ 2505].)

Reference

ReadOnlySymbolCollection.T. Class [▶ 2522]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.9.3.1 ReadOnlySymbolCollection.T..Empty Method

Returns an empty collection.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ReadOnlySymbolCollection<T> Empty()
```

Return Value

Type: ReadOnlySymbolCollection [▶ 2522].T [▶ 2522].
ReadOnlySymbolCollection<T>.

Reference

ReadOnlySymbolCollection.T. Class [▶ 2522]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10 SymbolCollection.T. Class

Interface represents a collection of ISymbol [▶ 2176] objects.

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.InstanceCollection [▶ 2460].T.
    TwinCAT.TypeSystem.Generic.SymbolCollection.T.
      TwinCAT.TypeSystem.SymbolCollection [▶ 2396]
Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SymbolCollection<T> : InstanceCollection<T>,
    ISymbolCollection<T>, IInstanceCollection<T>, IList<T>, ICollection<T>,
    IEnumerable<T>, IEnumerable
where T : class, ISymbol
```

Type Parameters

T

The SymbolCollection.T. type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the IList of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2530]	Returns a Read only version of this collection (shallow copy).
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2530]	Clones this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2531]	Creates an Empty SymbolCollection.T.
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstances [▶ 2531]	Try to get instances with predicate function

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.1 SymbolCollection.T. Properties

The SymbolCollection.T. [▶ 2526] generic type exposes the following members.

Properties

	Name	Description
	Count [▶ 2464]	Gets the collection count. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerList [▶ 2465]	Gets the IList of instances. (Inherited from InstanceCollection.T. [▶ 2460].)
	InnerPathDict [▶ 2465]	The Path dictionary (Inherited from InstanceCollection.T. [▶ 2460].)
	IsReadOnly [▶ 2465]	Gets a value indicating whether this instance is read only. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.Int32. [▶ 2466]	Gets or sets the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	Item.String. [▶ 2467]	Gets the IInstance [▶ 2052] with the specified instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	Mode [▶ 2468]	The mode this InstanceCollection.T. [▶ 2460] is working in. (Inherited from InstanceCollection.T. [▶ 2460].)

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.2 SymbolCollection.T. Methods

The SymbolCollection.T. [▶ 2526] generic type exposes the following members.

Methods

	Name	Description
	Add [▶ 2469]	Adds the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	AddRange [▶ 2470]	Adds the specified items to this collection. (Inherited from InstanceCollection.T. [▶ 2460].)
	AsReadOnly [▶ 2530]	Returns a Read only version of this collection (shallow copy).
	Clear [▶ 2471]	Clears this instance. (Inherited from InstanceCollection.T. [▶ 2460].)
	Clone [▶ 2530]	Clones this instance.
	Contains(String) [▶ 2472]	Determines whether this collection contains an IInstance [▶ 2052] with the specified InstanceName / InstancePath (Inherited from InstanceCollection.T. [▶ 2460].)
	Contains(T) [▶ 2472]	Determines whether this collection contains the specified IInstance [▶ 2052] (Inherited from InstanceCollection.T. [▶ 2460].)
	ContainsName [▶ 2473]	Determines whether the specified instance name contains name. (Inherited from InstanceCollection.T. [▶ 2460].)
	CopyTo [▶ 2474]	Copies this InstanceCollection.T. [▶ 2460] to the specified array. (Inherited from InstanceCollection.T. [▶ 2460].)
S	Empty [▶ 2531]	Creates an Empty SymbolCollection.T. [▶ 2526]
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2474]	Gets the enumerator. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetInstance [▶ 2475]	Gets the IInstance [▶ 2052]by instance path. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetInstanceByName [▶ 2476]	Gets the name of the instance by. (Inherited from InstanceCollection.T. [▶ 2460].)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	IndexOf [▶ 2476]	Determines the index of the specified IInstance [▶ 2052]. (Inherited from InstanceCollection.T. [▶ 2460].)
	Insert [▶ 2477]	Inserts the specified IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	Remove [▶ 2478]	Removes the specified item. (Inherited from InstanceCollection.T. [▶ 2460].)
	RemoveAt [▶ 2479]	Removes the IInstance [▶ 2052] at the specified index. (Inherited from InstanceCollection.T. [▶ 2460].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	TryGetInstance [▶ 2479]	Tries to get the IInstance [▶ 2052]. of the specified path. (Inherited from InstanceCollection.T. [▶ 2460].)

	Name	Description
	TryGetInstanceByName [▶ 2480]	Tries to get Instances by name. (Inherited from InstanceCollection.T. [▶ 2460].)
	TryGetInstances [▶ 2531]	Try to get instances with predicate function

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.2.1 SymbolCollection.T..AsReadOnly Method

Returns a Read only version of this collection (shallow copy).

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ReadOnlySymbolCollection<T> AsReadOnly()
```

Return Value

Type: ReadOnlySymbolCollection [▶ 2522].T [▶ 2526].
ReadOnlySymbolCollection<T>.

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.2.2 SymbolCollection.T..Clone Method

Clones this instance.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolCollection<T> Clone()
```

Return Value

Type: SymbolCollection [▶ 2526].T [▶ 2526].
SymbolCollection<T>.

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.2.3 SymbolCollection.T..Empty Method

Creates an Empty SymbolCollection.T. [▶ 2526]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static SymbolCollection<T> Empty()
```

Return Value

Type: SymbolCollection [▶ 2526].T [▶ 2526].
SymbolCollection<T>.

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.10.2.4 SymbolCollection.T..TryGetInstances Method

Try to get instances with predicate function

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool TryGetInstances(
    Func<T, bool> predicate,
    bool recurse,
    out IList<T> instances
)
```

Parameters

predicate	Type: System.Func.T [▶ 2526], Boolean. The predicate function
recurse	Type: System.Boolean if set to true the symbol hierarchy will be searched recursively.
instances	Type: System.Collections.Generic.IList.T [▶ 2526].. The instances.

Return Value

Type: Boolean
true if XXXX, false otherwise.

Reference

SymbolCollection.T. Class [▶ 2526]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.11 SymbolIterationMask Enumeration

Mask Flagset to specify filters for SymbolIterator.T. [▶ 2532].

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum SymbolIterationMask
```

Members

	Member name	Value	Description
	None	0	Uninitialized / None
	Structures	1	Iterates over Subelements of Structs
	Arrays	2	Iterates over Elements of Arrays
	Unions	4	Iterates over Subelements of Unions
	Pointer	8	Iterates over Pointer SubElements
	References	16	Iterates over References
	All	31	Iterates over All Complex/Combined types

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12 SymbolIterator.T. Class

Iterator class for enumerations of Symbols [▶ 2176].

Inheritance Hierarchy

System.Object
  TwinCAT.TypeSystem.Generic.SymbolIterator.T.
    TwinCAT.Ads.TypeSystem.SymbolIterator [▶ 1517]
Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class SymbolIterator<T> : IEnumerable<T>,
    IEnumerable
where T : class, ISymbol
```

Type Parameters

T	Concrete ISymbol [▶ 2176] type.

The SymbolIterator.T. type exposes the following members.

Constructors

	Name	Description
	SymbolIterator.T. (IInstanceCollection.T.) [▶ 2535]	Initializes a new instance of the SymbolIterator.T. class.
	SymbolIterator.T. (IEnumerable.T., Boolean) [▶ 2536]	Initializes a new instance of the SymbolIterator.T. class.
	SymbolIterator.T. (IInstanceCollection.T., Func.T, Boolean.) [▶ 2536]	Initializes a new instance of the SymbolIterator.T. class.
	SymbolIterator.T. (IEnumerable.T., Boolean, Func.T, Boolean.) [▶ 2537]	Initializes a new instance of the SymbolIterator.T. class.

Properties

	Name	Description
	Mask [▶ 2538]	Gets or sets the SymbolIterationMask [▶ 2532]
	SymbolRecursionDetection [▶ 2538]	Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default).

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2539]	Gets the enumerator that enumerates through a collection
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This iterator class can be used to iterate over collections of symbol trees (root symbols + sub symbols). By constructor the user can choose if the iterator works recursively within the symbol tree and optionally a filter function to select only specific symbols (predicate).

Examples

The following example shows how to determine, browse and filter symbols.

Browsing and filtering Symbols

```
using (AdsClient client = new AdsClient())
{
    CancellationToken cancel = CancellationToken.None;

    uint valueToRead = 0;
    uint valueToWrite = 42;

    client.Connect(AmsNetId.Local, 851);

    // Load all Symbols + DataTypes
    ISymbolLoader loader = SymbolLoaderFactory.Create(client, SymbolLoaderSettings.Default);

    ResultSymbols resultSymbols  = await loader.GetSymbolsAsync(cancel);

    if (resultSymbols.Succeeded)
    {
    Symbol symbol = (Symbol)resultSymbols.Symbols["MAIN.nCounter"];

    // Works for ALL Primitive 'ANY TYPES' Symbols
    ResultWriteAccess resultWrite = await symbol.WriteValueAsync(valueToWrite, cancel);
    ResultReadValueAccess resultRead = await symbol.ReadValueAsync(cancel);

    if (resultRead.Succeeded)
        valueToRead = (uint)resultRead.Value;

    // Simple filtering of Symbols
    Regex filterExpression = new Regex(pattern: @"^MAIN.*"); // Everything that starts with "MAIN"

    // FilterFunction that filters for the InstancePath
    Func<ISymbol, bool> filter = s => filterExpression.IsMatch(s.InstancePath);
    SymbolIterator iterator = new SymbolIterator(symbols: resultSymbols.Symbols, recurse: true, pred
icate: filter);

    foreach (ISymbol filteredSymbol in iterator)
    {
        Console.WriteLine(filteredSymbol.InstancePath);
    }
    }
}
```

Reference

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

System.Collections.Generic.IEnumerable.T.

6.12.12.1 SymbolIterator.T. Constructor

Overload List

	Name	Description
	SymbolIterator.T. (IInstanceCollection. T.) [▶ 2535]	Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.
	SymbolIterator.T. (IEnumerable.T., Boolean) [▶ 2536]	Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.
	SymbolIterator.T. (IInstanceCollection. T., Func.T, Boolean.) [▶ 2536]	Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.
	SymbolIterator.T. (IEnumerable.T., Boolean, Func.T, Boolean.) [▶ 2537]	Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.1.1 SymbolIterator.T. Constructor (IInstanceCollection.T.)

Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IInstanceCollection<T> symbols
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].T [▶ 2532]. The root symbols.

Reference

SymbolIterator.T. Class [▶ 2532]

SymbolIterator.T. Overload [▶ 2535]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.1.2 SymbolIterator.T. Constructor (IEnumerable.T., Boolean)

Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IEnumerable<T> symbols,
    bool recurse
)
```

Parameters

symbols	Type: System.Collections.Generic.IEnumerable.T [▶ 2532]. The root collection
recurse	Type: System.Boolean if set to true, the iterator works recursively over all subsymbols.

Reference

SymbolIterator.T. Class [▶ 2532]

SymbolIterator.T. Overload [▶ 2535]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.1.3 SymbolIterator.T. Constructor (IInstanceCollection.T., Func.T, Boolean.)

Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IInstanceCollection<T> symbols,
    Func<T, bool> predicate
)
```

Parameters

symbols	Type: TwinCAT.TypeSystem.IInstanceCollection [▶ 2057].T [▶ 2532]. The root symbols.
predicate	Type: System.Func.T [▶ 2532], Boolean. The predicate.

Exceptions

Exception	Condition
NotSupportedException	

Reference

SymbolIterator.T. Class [▶ 2532]

SymbolIterator.T. Overload [▶ 2535]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.1.4 SymbolIterator.T. Constructor (IEnumerable.T., Boolean, Func.T, Boolean.)

Initializes a new instance of the SymbolIterator.T. [▶ 2532] class.

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterator(
    IEnumerable<T> symbols,
    bool recurse,
    Func<T, bool> predicate
)
```

Parameters

symbols	Type: System.Collections.Generic.IEnumerable.T [▶ 2532]. Input collection (root objects).
recurse	Type: System.Boolean if set to true, the iterator works recursively over all subsymbols.
predicate	Type: System.Func.T [▶ 2532], Boolean. The predicate.

Reference

SymbolIterator.T. Class [▶ 2532]

SymbolIterator.T. Overload [▶ 2535]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.2 SymbolIterator.T. Properties

The SymbolIterator.T. [▶ 2532] generic type exposes the following members.

Properties

	Name	Description
	Mask [▶ 2538]	Gets or sets the SymbolIterationMask [▶ 2532]

	Name	Description
	SymbolRecursionDetection [▶ 2538]	Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default).

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.2.1 SymbolIterator.T..Mask Property

Gets or sets the SymbolIterationMask [▶ 2532]

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public SymbolIterationMask Mask { get; set; }
```

Property Value

Type: SymbolIterationMask [▶ 2532]
The mask.

Remarks

This property can be used for prefiltering the iterator without using a predicate function.

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.2.2 SymbolIterator.T..SymbolRecursionDetection Property

Gets or sets a value indicating whether the iterator checks for Symbol recursions (true by default).

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool SymbolRecursionDetection { get; set; }
```

Property Value

Type: Boolean
true if recursion checking, false switched off check.

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.3 SymbolIterator.T. Methods

The SymbolIterator.T. [▶ 2532] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetEnumerator [▶ 2539]	Gets the enumerator that enumerates through a collection
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.12.12.3.1 SymbolIterator.T..GetEnumerator Method

Gets the enumerator that enumerates through a collection

Namespace: TwinCAT.TypeSystem.Generic [▶ 2441]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public IEnumerator<T> GetEnumerator()
```

Return Value

Type: IEnumerator.T [▶ 2532].
A IEnumerator.T. that can be used to iterate through the collection.

Implements

IEnumerable.T..GetEnumerator.

Reference

SymbolIterator.T. Class [▶ 2532]

TwinCAT.TypeSystem.Generic Namespace [▶ 2441]

6.13 TwinCAT.ValueAccess Namespace

Namespace for the common (non ADS dependent) value access.

Classes

	Class	Description
	CannotAccessValueException [▶ 2541]	Class CannotAccessValueException. This class cannot be inherited. Implements the SymbolException [▶ 2401]
	ResultAccess [▶ 2556]	Result class for an asynchronous access operation.
	ResultReadDynamicValueAccess [▶ 2562]	Asynchronous read access result returning an Dynamic value object (IDynamicValue [▶ 2011]. Implements the ResultReadValueAccess.T. [▶ 2568]
	ResultReadRawAccess [▶ 2564]	Asynchronous read access result object, reading raw (byte[]) data into memory locations. Implements the ResultReadValueAccess.T. [▶ 2568]
	ResultReadValueAccess [▶ 2566]	Asynchronous read access result returning an untyped Value object type 'object'. Implements the ResultAccess [▶ 2556]
	ResultReadValueAccess.T. [▶ 2568]	Result object of an asynchronous read of a specific value of type T. Implements the ResultAccess [▶ 2556]
	ResultRpcMethodAccess [▶ 2571]	Asynchronous Invoke RPC Method result class. Implements the ResultReadValueAccess [▶ 2566]
	ResultWriteAccess [▶ 2575]	Asynchronous write access result class.

Interfaces

	Interface	Description
	IAccessorRawValue [▶ 2544]	Helper Interface to access Symbol Values as byte Arrays
	IAccessorValueFactory [▶ 2552]	Factory interfaces for Accessor implementations.

Enumerations

	Enumeration	Description
	SymbolNotificationTypes [▶ 2579]	Specifies the Notification type of ADS Notifications
	ValueCreationModes [▶ 2580]	Creation mode for Values

6.13.1 CannotAccessValueException Class

Class CannotAccessValueException. This class cannot be inherited. Implements the SymbolException [▶ 2401]

Inheritance Hierarchy

System.Object
  System.Exception
    TwinCAT.AdsException [▶ 57]
      TwinCAT.TypeSystem.SymbolException [▶ 2401]
        TwinCAT.ValueAccess.CannotAccessValueException

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[SerializableAttribute]
public sealed class CannotAccessValueException : SymbolException
```

The CannotAccessValueException type exposes the following members.

Constructors

	Name	Description
	CannotAccessValueException. [▶ 2542]	Initializes a new instance of the CannotAccessValueException class.
	CannotAccessValueException(ISymbol) [▶ 2543]	Initializes a new instance of the CannotAccessValueException class.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)

	Name	Description
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.TypeSystem.SymbolException [▶ 2401]

6.13.1.1 CannotAccessValueException Constructor

Overload List

	Name	Description
	CannotAccessValueException. [▶ 2542]	Initializes a new instance of the CannotAccessValueException [▶ 2541] class.
	CannotAccessValueException(ISymbol) [▶ 2543]	Initializes a new instance of the CannotAccessValueException [▶ 2541] class.

Reference

CannotAccessValueException Class [▶ 2541]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.1.1.1 CannotAccessValueException Constructor

Initializes a new instance of the CannotAccessValueException [▶ 2541] class.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public CannotAccessValueException()
```

Reference

CannotAccessValueException Class [▶ 2541]

CannotAccessValueException Overload [▶ 2542]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.1.1.2 CannotAccessValueException Constructor (ISymbol)

Initializes a new instance of the CannotAccessValueException [▶ 2541] class.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads (in TwinCAT.Ads.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public CannotAccessValueException(
    ISymbol symbol
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176] The symbol.

Reference

CannotAccessValueException Class [▶ 2541]

CannotAccessValueException Overload [▶ 2542]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.1.2 CannotAccessValueException Properties

The CannotAccessValueException [▶ 2541] type exposes the following members.

Properties

	Name	Description
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception.)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception.)

	Name	Description
	InstancePath [▶ 2410]	Gets the instance path. (Inherited from SymbolException [▶ 2401].)
	Message	Gets a message that describes the current exception. (Inherited from Exception.)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)
	StackTrace	Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)
	Symbol [▶ 2411]	Gets the symbol. (Inherited from SymbolException [▶ 2401].)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception.)

Reference

CannotAccessValueException Class [▶ 2541]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.1.3 CannotAccessValueException Methods

The CannotAccessValueException [▶ 2541] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	GetBaseException	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetObjectData [▶ 2412]	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from SymbolException [▶ 2401].)
	GetType	Gets the runtime type of the current instance. (Inherited from Exception.)
	ToString	Creates and returns a string representation of the current exception. (Inherited from Exception.)

Reference

CannotAccessValueException Class [▶ 2541]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2 IAccessorRawValue Interface

Helper Interface to access Symbol Values as byte Arrays

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAccessorRawValue
```

The IAccessorRawValue type exposes the following members.

Properties

	Name	Description
	DefaultValueEncoding [▶ 2546]	Gets the default value encoding.
	ValueFactory [▶ 2546]	Gets the value factory

Methods

	Name	Description
	ReadArrayElementRawAsync [▶ 2547]	
	ReadRawAsync [▶ 2548]	
	TryReadArrayElementRaw [▶ 2548]	
	TryReadRaw [▶ 2549]	
	TryWriteArrayElementRaw [▶ 2550]	
	TryWriteRaw [▶ 2550]	
	WriteArrayElementRawAsync [▶ 2551]	
	WriteRawAsync [▶ 2552]	

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.1 IAccessorRawValue Properties

The IAccessorRawValue [▶ 2544] type exposes the following members.

Properties

	Name	Description
	DefaultValueEncoding [▶ 2546]	Gets the default value encoding.

	Name	Description
	ValueFactory [▶ 2546]	Gets the value factory

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.1.1 IAccessorRawValue.DefaultValueEncoding Property

Gets the default value encoding.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Encoding DefaultValueEncoding { get; }
```

Property Value

Type: Encoding
The default value encoding.

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.1.2 IAccessorRawValue.ValueFactory Property

Gets the value factory

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
IAccessorValueFactory ValueFactory { get; }
```

Property Value

Type: IAccessorValueFactory [▶ 2552]
The value factory or null if only Raw Values only.

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2 IAccessorRawValue Methods

The IAccessorRawValue [▶ 2544] type exposes the following members.

Methods

	Name	Description
	ReadArrayElementRawAsync [▶ 2547]	
	ReadRawAsync [▶ 2548]	
	TryReadArrayElementRaw [▶ 2548]	
	TryReadRaw [▶ 2549]	
	TryWriteArrayElementRaw [▶ 2550]	
	TryWriteRaw [▶ 2550]	
	WriteArrayElementRawAsync [▶ 2551]	
	WriteRawAsync [▶ 2552]	

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.1 IAccessorRawValue.ReadArrayElementRawAsync Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadRawAccess> ReadArrayElementRawAsync(
    IArrayInstance arrayInstance,
    int[] indices,
    Memory readBuffer,
    void cancel
)
```

Parameters

arrayInstance	Type: TwinCAT.TypeSystem.IArrayInstance [▶ 1964]
indices	Type: .System.Int32.

readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.2 IAccessorRawValue.ReadRawAsync Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultReadRawAccess> ReadRawAsync(
    ISymbol symbolInstance,
    Memory readBuffer,
    void cancel
)
```

Parameters

symbolInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
readBuffer	Type: Memory
cancel	Type: System.Void

Return Value

Type: Task.ResultReadRawAccess [▶ 2564].

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.3 IAccessorRawValue.TryReadArrayElementRaw Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryReadArrayElementRaw(
    IArrayInstance arrayInstance,
    int[] indices,
    Memory readBuffer,
    void timeStamp
)
```

Parameters

arrayInstance	Type: TwinCAT.TypeSystem.IArrayInstance [▶ 1964]
indices	Type: .System.Int32.
readBuffer	Type: Memory
timeStamp	Type: System.Void

Return Value

Type: Int32

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.4 IAccessorRawValue.TryReadRaw Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryReadRaw(
    ISymbol symbolInstance,
    Memory readBuffer,
    void timeStamp
)
```

Parameters

symbolInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
readBuffer	Type: Memory
timeStamp	Type: System.Void

Return Value

Type: Int32

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.5 IAccessorRawValue.TryWriteArrayElementRaw Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryWriteArrayElementRaw(
    IArrayInstance arrayInstance,
    int[] indices,
    ReadOnlyMemory writeBuffer,
    void timeStamp
)
```

Parameters

arrayInstance	Type: TwinCAT.TypeSystem.IArrayInstance [▶ 1964]
indices	Type: .System.Int32.
writeBuffer	Type: ReadOnlyMemory
timeStamp	Type: System.Void

Return Value

Type: Int32

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.6 IAccessorRawValue.TryWriteRaw Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
int TryWriteRaw(
    ISymbol symbolInstance,
    ReadOnlyMemory writeBuffer,
    void timeStamp
)
```

Parameters

symbolInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
writeBuffer	Type: ReadOnlyMemory
timeStamp	Type: System.Void

Return Value

Type: Int32

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.7 IAccessorRawValue.WriteArrayElementRawAsync Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWriteAccess> WriteArrayElementRawAsync(
    IArrayInstance arrayInstance,
    int[] indices,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

arrayInstance	Type: TwinCAT.TypeSystem.IArrayInstance [▶ 1964]
indices	Type: .System.Int32.
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWriteAccess [▶ 2575].

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.2.2.8 IAccessorRawValue.WriteRawAsync Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Task<ResultWriteAccess> WriteRawAsync(
    ISymbol symbolInstance,
    ReadOnlyMemory writeBuffer,
    void cancel
)
```

Parameters

symbolInstance	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
writeBuffer	Type: ReadOnlyMemory
cancel	Type: System.Void

Return Value

Type: Task.ResultWriteAccess [▶ 2575].

Reference

IAccessorRawValue Interface [▶ 2544]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.3 IAccessorValueFactory Interface

Factory interfaces for Accessor implementations.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public interface IAccessorValueFactory
```

The IAccessorValueFactory type exposes the following members.

Methods

	Name	Description
	CreatePrimitiveValue [▶ 2553]	

	Name	Description
	CreateValue(ISymbol, ReadOnlyMemory, Void) [▶ 2554]	
	CreateValue(ISymbol, ReadOnlyMemory, Void, Byte) [▶ 2555]	

Remarks

This interface is used by a custom Accessor class to create Value objects from memory representations.

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.3.1 IAccessorValueFactory Methods

The IAccessorValueFactory [▶ 2552] type exposes the following members.

Methods

	Name	Description
	CreatePrimitiveValue [▶ 2553]	
	CreateValue(ISymbol, ReadOnlyMemory, Void) [▶ 2554]	
	CreateValue(ISymbol, ReadOnlyMemory, Void, Byte) [▶ 2555]	

Reference

IAccessorValueFactory Interface [▶ 2552]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.3.1.1 IAccessorValueFactory.CreatePrimitiveValue Method

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object CreatePrimitiveValue(
    ISymbol symbol,
    ReadOnlyMemory sourceData,
    void timeStamp
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
sourceData	Type: ReadOnlyMemory
timeStamp	Type: System.Void

Return Value

Type: Object

Reference

IAccessorValueFactory Interface [▶ 2552]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.3.1.2 IAccessorValueFactory.CreateValue Method

Overload List

	Name	Description
	CreateValue(ISymbol, ReadOnlyMemory, Void) [▶ 2554]	
	CreateValue(ISymbol, ReadOnlyMemory, Void, Byte) [▶ 2555]	

Reference

IAccessorValueFactory Interface [▶ 2552]

TwinCAT.ValueAccess Namespace [▶ 2540]

IAccessorValueFactory.CreateValue Method (ISymbol, ReadOnlyMemory`1, Void)

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object CreateValue(
    ISymbol symbol,
    ReadOnlyMemory sourceData,
    void timeStamp
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
sourceData	Type: ReadOnlyMemory
timeStamp	Type: System.Void

Return Value

Type: Object

Reference

IAccessorValueFactory Interface [▶ 2552]

CreateValue Overload [▶ 2554]

TwinCAT.ValueAccess Namespace [▶ 2540]

IAccessorValueFactory.CreateValue Method (ISymbol, ReadOnlyMemory`1, Void, Byte)

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
Object CreateValue(
    ISymbol symbol,
    ReadOnlyMemory sourceData,
    void parent,
    byte timeStamp
)
```

Parameters

symbol	Type: TwinCAT.TypeSystem.ISymbol [▶ 2176]
sourceData	Type: ReadOnlyMemory
parent	Type: System.Void
timeStamp	Type: System.Byte

Return Value

Type: Object

Reference

IAccessorValueFactory Interface [▶ 2552]

CreateValue Overload [▶ 2554]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4 ResultAccess Class

Result class for an asynchronous access operation.

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess
    TwinCAT.ValueAccess.ResultReadValueAccess.T. [▶ 2568]
    TwinCAT.ValueAccess.ResultWriteAccess [▶ 2575]

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultAccess
```

The ResultAccess type exposes the following members.

Constructors

	Name	Description
	ResultAccess(Int32) [▶ 2557]	Initializes a new instance of the ResultAccess class.
	ResultAccess(Int32, DateTimeOffset) [▶ 2558]	Initializes a new instance of the ResultReadValueAccess [▶ 2566] struct.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess represents a failed access.
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess represents a succeeded access.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess object.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Remarks

This result is independent of any used protocol and is used by the different protocol providers (ADS, OPC, IOT etc.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.1 ResultAccess Constructor

Overload List

	Name	Description
	ResultAccess(Int32) [▶ 2557]	Initializes a new instance of the ResultAccess [▶ 2556] class.
	ResultAccess(Int32, DateTimeOffset) [▶ 2558]	Initializes a new instance of the ResultReadValueAccess [▶ 2566] struct.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.1.1 ResultAccess Constructor (Int32)

Initializes a new instance of the ResultAccess [▶ 2556] class.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultAccess(
    int errorResult
)
```

Parameters

errorResult	Type: System.Int32 The result code of the communication access..

Reference

ResultAccess Class [▶ 2556]

ResultAccess Overload [▶ 2557]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.1.2 ResultAccess Constructor (Int32, DateTimeOffset)

Initializes a new instance of the ResultReadValueAccess [▶ 2566] struct.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultAccess(
    int errorResult,
    DateTimeOffset time
)
```

Parameters

errorResult	Type: System.Int32 The result code of the operation.
time	Type: System.DateTimeOffset The timestamp of the operation.

Reference

ResultAccess Class [▶ 2556]

ResultAccess Overload [▶ 2557]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.2 ResultAccess Properties

The ResultAccess [▶ 2556] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access.
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.2.1 ResultAccess.DateTime Property

The Timestamp / the date time of the communication access.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public DateTimeOffset DateTime { get; }
```

Property Value

Type: DateTimeOffset

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.2.2 ResultAccess.ErrorCode Property

The ErrorCode of the communication access

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public int ErrorCode { get; }
```

Property Value

Type: Int32

Remarks

A value of '0' represents success.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.2.3 ResultAccess.Failed Property

Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Failed { get; }
```

Property Value

Type: Boolean
true if failed; otherwise, false.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.2.4 ResultAccess.Succeeded Property

Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public bool Succeeded { get; }
```

Property Value

Type: Boolean
true if succeeded; otherwise, false.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.3 ResultAccess Methods

The ResultAccess [▶ 2556] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object.
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.4.3.1 ResultAccess.SetError Method

Sets the error to this ResultAccess [▶ 2556] object.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public void SetError(
    int errorCode
)
```

Parameters

errorCode	Type: System.Int32 The error code.

Reference

ResultAccess Class [▶ 2556]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.5 ResultReadDynamicValueAccess Class

Asynchronous read access result returning an Dynamic value object (IDynamicValue [▶ 2011]. Implements the ResultReadValueAccess.T. [▶ 2568]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2568].IDynamicValue [▶ 2011].
      TwinCAT.ValueAccess.ResultReadDynamicValueAccess

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadDynamicValueAccess : ResultReadValueAccess<IDynamicValue>
```

The ResultReadDynamicValueAccess type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.ValueAccess.ResultReadValueAccess.T. [▶ 2568]

6.13.5.1 ResultReadDynamicValueAccess Properties

The ResultReadDynamicValueAccess [▶ 2562] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Reference

ResultReadDynamicValueAccess Class [▶ 2562]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.5.2 ResultReadDynamicValueAccess Methods

The ResultReadDynamicValueAccess [▶ 2562] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadDynamicValueAccess Class [▶ 2562]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.6 ResultReadRawAccess Class

Asynchronous read access result object, reading raw (byte[]) data into memory locations. Implements the ResultReadValueAccess.T. [▶ 2568]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2568].Memory.
      TwinCAT.ValueAccess.ResultReadRawAccess

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadRawAccess : ResultReadValueAccess<Memory>
```

The ResultReadRawAccess type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
S	Empty [▶ 2565]	Gets the empty ResultAnyValue [▶ 998] object.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)

	Name	Description
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.ValueAccess.ResultReadValueAccess.T. [▶ 2568]

6.13.6.1 ResultReadRawAccess Properties

The ResultReadRawAccess [▶ 2564] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	Empty [▶ 2565]	Gets the empty ResultAnyValue [▶ 998] object.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Reference

ResultReadRawAccess Class [▶ 2564]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.6.1.1 ResultReadRawAccess.Empty Property

Gets the empty ResultAnyValue [▶ 998] object.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultReadRawAccess Empty { get; }
```

Property Value

Type: ResultReadRawAccess [▶ 2564]
The empty / unprocessed result.

Reference

ResultReadRawAccess Class [▶ 2564]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.6.2 ResultReadRawAccess Methods

The ResultReadRawAccess [▶ 2564] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadRawAccess Class [▶ 2564]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.7 ResultReadValueAccess Class

Asynchronous read access result returning an untyped Value object type 'object'. Implements the ResultAccess [▶ 2556]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2568].Object.
      TwinCAT.ValueAccess.ResultReadValueAccess
        TwinCAT.ValueAccess.ResultRpcMethodAccess [▶ 2571]

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadValueAccess : ResultReadValueAccess<Object>
```

The ResultReadValueAccess type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.ValueAccess.ResultAccess [▶ 2556]

6.13.7.1 ResultReadValueAccess Properties

The ResultReadValueAccess [▶ 2566] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)

	Name	Description
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Reference

ResultReadValueAccess Class [▶ 2566]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.7.2 ResultReadValueAccess Methods

The ResultReadValueAccess [▶ 2566] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadValueAccess Class [▶ 2566]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.8 ResultReadValueAccess.T. Class

Result object of an asynchronous read of a specific value of type T. Implements the ResultAccess [▶ 2556]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultReadValueAccess.T.
      TwinCAT.ValueAccess.ResultReadDynamicValueAccess [▶ 2562]

TwinCAT.ValueAccess.ResultReadRawAccess [▶ 2564]
TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2566]

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultReadValueAccess<T> : ResultAccess
```

Type Parameters

T	The type of the value.

The ResultReadValueAccess.T. type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.ValueAccess.ResultAccess [▶ 2556]

6.13.8.1 ResultReadValueAccess.T. Properties

The ResultReadValueAccess.T. [▶ 2568] generic type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value

Reference

ResultReadValueAccess.T. Class [▶ 2568]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.8.1.1 ResultReadValueAccess.T..Value Property

The value

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public T Value { get; }
```

Property Value

Type: T [▶ 2568]

Reference

ResultReadValueAccess.T. Class [▶ 2568]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.8.2 ResultReadValueAccess.T. Methods

The ResultReadValueAccess.T. [▶ 2568] generic type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultReadValueAccess.T. Class [▶ 2568]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.9 ResultRpcMethodAccess Class

Asynchronous Invoke RPC Method result class. Implements the ResultReadValueAccess [▶ 2566]

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2568].Object.
      TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2566]
        TwinCAT.ValueAccess.ResultRpcMethodAccess

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultRpcMethodAccess : ResultReadValueAccess
```

The ResultRpcMethodAccess type exposes the following members.

Constructors

	Name	Description
	ResultRpcMethodAccess [▶ 2572]	Initializes a new instance of the ResultRpcMethod [▶ 1025] struct.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	OutParameters [▶ 2574]	Gets the out parameters.
	ReturnValue [▶ 2574]	Gets the return value of the RpcMethod (optionally).
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

TwinCAT.ValueAccess.ResultReadValueAccess [▶ 2566]

6.13.9.1 ResultRpcMethodAccess Constructor

Initializes a new instance of the ResultRpcMethod [▶ 1025] struct.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultRpcMethodAccess(
    Object returnValue,
    Object[] outParameters,
    int errorCode,
    DateTimeOffset timeStamp
)
```

Parameters

returnValue	Type: System.Object The value.
outParameters	Type: .System.Object. The out parameters.
errorCode	Type: System.Int32 The error code.
timeStamp	Type: System.DateTimeOffset The time stamp.

Reference

ResultRpcMethodAccess Class [▶ 2571]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.9.2 ResultRpcMethodAccess Properties

The ResultRpcMethodAccess [▶ 2571] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	OutParameters [▶ 2574]	Gets the out parameters.
	ReturnValue [▶ 2574]	Gets the return value of the RpcMethod (optionally).
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)
	Value [▶ 2570]	The value (Inherited from ResultReadValueAccess.T. [▶ 2568].)

Reference

ResultRpcMethodAccess Class [▶ 2571]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.9.2.1 ResultRpcMethodAccess.OutParameters Property

Gets the out parameters.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object[] OutParameters { get; }
```

Property Value

Type: .Object.
The out parameters.

Reference

ResultRpcMethodAccess Class [▶ 2571]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.9.2.2 ResultRpcMethodAccess.ReturnValue Property

Gets the return value of the RpcMethod (optionally).

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public Object ReturnValue { get; }
```

Property Value

Type: Object
The return value or NULL

Reference

ResultRpcMethodAccess Class [▶ 2571]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.9.3 ResultRpcMethodAccess Methods

The ResultRpcMethodAccess [▶ 2571] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)

	Name	Description
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultRpcMethodAccess Class [▶ 2571]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10 ResultWriteAccess Class

Asynchronous write access result class.

Inheritance Hierarchy

System.Object
  TwinCAT.ValueAccess.ResultAccess [▶ 2556]
    TwinCAT.ValueAccess.ResultWriteAccess
Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public class ResultWriteAccess : ResultAccess
```

The ResultWriteAccess type exposes the following members.

Constructors

	Name	Description
	ResultWriteAccess(Int32) [▶ 2577]	Initializes a new instance of the ResultWriteAccess class.
	ResultWriteAccess(Int32, DateTimeOffset) [▶ 2577]	Initializes a new instance of the ResultWriteAccess struct.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
S	Empty [▶ 2578]	Gets the empty ResultWriteAccess object.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.1 ResultWriteAccess Constructor

Overload List

	Name	Description
	ResultWriteAccess(Int32) [▶ 2577]	Initializes a new instance of the ResultWriteAccess [▶ 2575] class.
	ResultWriteAccess(Int32, DateTimeOffset) [▶ 2577]	Initializes a new instance of the ResultWriteAccess [▶ 2575] struct.

Reference

ResultWriteAccess Class [▶ 2575]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.1.1 ResultWriteAccess Constructor (Int32)

Initializes a new instance of the ResultWriteAccess [▶ 2575] class.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultWriteAccess(
    int errorResult
)
```

Parameters

errorResult	Type: System.Int32 The result code of the communication access..

Reference

ResultWriteAccess Class [▶ 2575]

ResultWriteAccess Overload [▶ 2576]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.1.2 ResultWriteAccess Constructor (Int32, DateTimeOffset)

Initializes a new instance of the ResultWriteAccess [▶ 2575] struct.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public ResultWriteAccess(
    int errorResult,
    DateTimeOffset timeStamp
)
```

Parameters

errorResult	Type: System.Int32 The error result.
timeStamp	Type: System.DateTimeOffset The time stamp.

Reference

ResultWriteAccess Class [▶ 2575]

ResultWriteAccess Overload [▶ 2576]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.2 ResultWriteAccess Properties

The ResultWriteAccess [▶ 2575] type exposes the following members.

Properties

	Name	Description
	DateTime [▶ 2559]	The Timestamp / the date time of the communication access. (Inherited from ResultAccess [▶ 2556].)
	Empty [▶ 2578]	Gets the empty ResultWriteAccess [▶ 2575] object.
	ErrorCode [▶ 2559]	The ErrorCode of the communication access (Inherited from ResultAccess [▶ 2556].)
	Failed [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a failed access. (Inherited from ResultAccess [▶ 2556].)
	Succeeded [▶ 2560]	Gets a value indicating whether this ResultAccess [▶ 2556] represents a succeeded access. (Inherited from ResultAccess [▶ 2556].)

Reference

ResultWriteAccess Class [▶ 2575]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.2.1 ResultWriteAccess.Empty Property

Gets the empty ResultWriteAccess [▶ 2575] object.

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
public static ResultWriteAccess Empty { get; }
```

Property Value

Type: ResultWriteAccess [▶ 2575]
The empty / unprocessed result.

Reference

ResultWriteAccess Class [▶ 2575]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.10.3 ResultWriteAccess Methods

The ResultWriteAccess [▶ 2575] type exposes the following members.

Methods

	Name	Description
	Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as the default hash function. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	SetError [▶ 2561]	Sets the error to this ResultAccess [▶ 2556] object. (Inherited from ResultAccess [▶ 2556].)
	ToString	Returns a string that represents the current object. (Inherited from Object.)

Reference

ResultWriteAccess Class [▶ 2575]

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.11 SymbolNotificationTypes Enumeration

Specifies the Notification type of ADS Notifications

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum SymbolNotificationTypes
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized
	Value	1	ValueChanged notifications
	RawValue	2	RawValueChanged notifications
	Both	3	ValueChanged + RawValueChanged notifications

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

6.13.12 ValueCreationModes Enumeration

Creation mode for Values

Namespace: TwinCAT.ValueAccess [▶ 2540]
Assembly: TwinCAT.Ads.Abstractions (in TwinCAT.Ads.Abstractions.dll) Version: 5.0.294+Branch.releases-5.0.Sha.90bb9a1b43b6095934fddca3e72bc0ea15da1c14

Syntax

C#

```
[FlagsAttribute]
public enum ValueCreationModes
```

Members

	Member name	Value	Description
	None	0	None / Uninitialized Mask
	Primitives	1	Convert to .NET Primitives, wherever possible.
	Enums	2	Use IEnumValue on EnumTypes instead of .NET Primitives
	FullDynamics	4	Wraps all Primitives also in IValue Objects
	PlcOpenTypes	8	Use PlcOpen Times (TIME, LTIME, DT, TOD, DATETIME) instead of .NET Primitives DateTime and TimeSpan
	Default	1	Default settings for the value creation mode (Translate to Primitives)

Remarks

This setting is used by the ValueFactory/ to create Read values. In default primitive mode all values will be transferred to Primitive .NET Symbols if possible. E.g PlcOpen.TIME --> TimeSpan, IEnumValue --> .NET Primitives. They won't be wrapped into

Reference

TwinCAT.ValueAccess Namespace [▶ 2540]

More Information:
www.beckhoff.com/tc1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com